

Oracle® Application Server

Administration Guide

Release 4.0.8.1

September 1999

Part No. A60172-03

ORACLE®

Oracle Application Server Release 4.0.8.1 Administration Guide

Part No. A60172-03

Copyright © 1996, 1999, Oracle Corporation. All rights reserved.

Primary Author: Sheryl Maring

Contributing Authors: Tom Van Raalte, Dave Stahler, Sanjay Singh, Janice Nygard, Pooja Sharma, Betsy Mooney

Contributors: Charlie Shapiro, Jun Wang, Seshu Adunuthula, Eric Deslauriers, Scott Christly, Ramani Jagadeba, Loc Nhan, Sajjad Rahman

The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the programs.

The programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these programs, no part of these programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle is a registered trademark, and the Oracle logo, NLS*WorkBench, Pro*COBOL, Pro*FORTRAN, Pro*Pascal, SQL*Loader, SQL*Module, SQL*Net, SQL*Plus, Oracle7, Oracle Server, Oracle Server Manager, Oracle Call Interface, Oracle7 Enterprise Backup Utility, Oracle TRACE, Oracle WebServer, Oracle Web Application Server, Oracle Application Server, Oracle Network Manager, Secure Network Services, Oracle Parallel Server, Advanced Replication Option, Oracle Data Query, Cooperative Server Technology, Oracle Toolkit, Oracle MultiProtocol Interchange, Oracle Names, Oracle Book, Pro*C, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Contents

Preface	xi
1 Overview	
Component Overview	1-1
Sample Path of a Request	1-2
Using Oracle Application Server	1-3
Oracle Application Server Welcome Page	1-4
Accessing the Welcome Page.....	1-5
Oracle Application Server Manager (OAS Manager)	1-6
Node Manager Operations and Icons.....	1-7
Reloading and Refreshing the Oracle Application Server.....	1-8
Oracle Application Server Utilities (OAS Utilities)	1-8
Install	1-9
Tuning	1-9
External Listener	1-9
Migrate	1-9
Log Analyzer	1-9
Database Browser	1-10
2 Managing your Site and Server	
Understanding your Web Site and Oracle Application Server	2-1
Understanding the OAS Manager Listeners	2-3
Node Manager Listener	2-3
Administration Utility Listener	2-4

Accessing the Oracle Application Server Manager	2-5
Configuring Nodes on your Web Site	2-6
Adding a New Node	2-7
Configuring Oracle Application Server Processes	2-9
Managing Oracle Application Server Processes	2-9
Monitoring your Oracle Application Server	2-11
Accessing the Oracle Application Server Monitoring Window	2-13

3 Managing and Configuring HTTP Listeners

Multiple Addresses and Ports	3-1
Duplicate Port Numbers and IP Addresses.....	3-2
How Oracle Application Server Manages Multiport Listeners	3-2
Administering Your HTTP Listeners	3-3
Monitoring Your HTTP Listeners	3-4
Accessing the HTTP Listeners Monitoring Window	3-5
Adding a New Listener	3-5
Starting the Listener	3-7
Configuring a Listener	3-7
Configuring Network Parameters.....	3-8
Configuring Server Parameters	3-13
Configuring User/Group Parameters	3-15
Configuring Directory Mappings.....	3-17
Configuring Language Extensions.....	3-19
Configuring MIME Types	3-20
Configuring Encoding Extensions.....	3-21
SSL CA Roots.....	3-22
Security	3-23
Using Third-Party Listeners	3-25

4 Registering (Migrating) Third-Party HTTP Servers

Overview	4-1
Registering Netscape Servers	4-1
To Manually Register Netscape Server Using the OAS Utility Page	4-2
Starting, Stopping, and Configuring the Netscape Server	4-3
Changes Made During Registration	4-3

Registering Microsoft Internet Information Servers (NT Only)	4-4
Enabling Microsoft Internet Information Server	4-5
Authentication	4-9
Registering Apache Servers (UNIX Only)	4-10
Apache adaptor Information	4-10
Files of Interest	4-11
Setting up Apache	4-11
Register Apache Server Using the OAS Utility Page	4-11
Starting, Stopping, and Configuring the Apache Server	4-14
Achieving Better Performance	4-14
Running the JCO Cartridge with the Apache Server	4-14
 5 Object Request Broker Administration	
Object Request Broker (ORB)	5-1
Configuring ORB General Parameters	5-2
Managing the ORB Components	5-4
Starting and Stopping the ORB	5-4
 6 Database Access Descriptors (DADs)	
What is a DAD?	6-1
Creating a DAD	6-2
DAD Name	6-4
Database User	6-5
Database User Password	6-5
Confirm Password	6-5
Database Location (Host)	6-5
Database Name (ORACLE_SID)	6-5
Connect String	6-6
Create Database User	6-6
Change Database User Password	6-6
Store the username and password in the DAD	6-6
Configuring a DAD	6-8
Adding Transactions	6-8
Adding Advanced Features to a DAD	6-11

7 Introduction to Applications

Cartridges and Applications	7-1
Cartridge-based Applications	7-2
IIOP Based Applications	7-3
Common Application Programming Models	7-6
The Request-Response Model	7-6
The Session Model	7-6
The Transaction Model	7-7
Tracing Requests for Applications	7-8
HTTP Requests	7-8
CORBA/IIOP Requests	7-8

8 Application Administration

Concepts	8-1
Cartridge-based Applications	8-2
ECO/Java and EJB Applications	8-3
Application Configuration Roadmap	8-6
Adding and Configuring Applications	8-7
Adding Applications	8-8
Configuring Applications	8-14
Configuration Forms	8-17
Configuring Cartridge Server Options	8-18
Configuring Hosts for Load Balancing	8-19
Configuring Logging Parameters	8-21
Configuring Web Parameters	8-22
Configuring Transactions	8-25
Configuring Environment Variables	8-25
Configuring the Java Environment	8-27
Managing Applications	8-27
Displaying Configuration Information	8-28
Monitoring Running Applications	8-29
Renaming Applications	8-31
Reloading Applications	8-31
Deleting Applications	8-32
Stopping Processes and Applications	8-33

Stopping and Restarting Listeners	8-34
---	------

9 Cartridge and Component Administration

Concepts	9-1
Cartridges	9-2
CORBA Components	9-2
Configuration Roadmap	9-3
Adding Cartridges	9-4
Adding a Cartridge While Adding an Application.....	9-5
Adding a CWeb Cartridge	9-7
Adding a C++ CORBA Cartridge	9-8
Adding a JServlet Cartridge.....	9-8
Adding a JWeb Cartridge	9-9
Adding a LiveHTML Cartridge.....	9-10
Adding a Perl Cartridge	9-12
Adding a PL/SQL Cartridge	9-13
Adding a Cartridge from a Configuration File	9-15
Adding CORBA Components	9-15
Configuring Cartridges	9-16
Displaying Cartridge Configuration Forms	9-16
Configuring Tuning Parameters	9-17
Configuring Virtual Paths	9-19
Configuring C Cartridge Parameters.....	9-22
Configuring JWeb Cartridge Parameters.....	9-24
Configuring LiveHTML Cartridge Parameters.....	9-24
Configuring ODBC Cartridge Parameters.....	9-27
Configuring Perl Cartridge Parameters	9-27
Configuring PL/SQL Cartridge Parameters	9-28
Configuring ECO/Java and EJB Components.....	9-30
Configuring Application Instances.....	9-30
Configuring CORBA Objects	9-34
Managing Cartridges.....	9-35
Monitoring Cartridges	9-37
Stopping Cartridges	9-39
Deleting Cartridges	9-39

Displaying Cartridge Information and Changing the Cartridge Display Name	9-40
Advanced Configuration Topics	9-41
Stateful Mode	9-41
Stateless Mode	9-41
Max Clients	9-42

10 Logging and Database Utilities

Configuring Logging Features	10-1
Accessing the Logging Forms	10-2
General Logging Configuration.....	10-2
Configuring System Logging.....	10-4
Configuring Attribute Logging	10-9
Configuring XLF Logging	10-14
Installing the Utilities	10-21
Before You Install.....	10-22
Task 1: Install the Logger Tables	10-22
Task 2: Install the PL/SQL Toolkit.....	10-23
Task 3: Install the Log Analyzer	10-25
Task 4: Install the Database Browser	10-26
Task 5: Grant Privileges to the DAD User.....	10-28
Using the Log Analyzer	10-28
Overview.....	10-28
Uploading Log Files into the Database.....	10-29
Viewing Log Entries	10-31
Running Log Reports	10-34
Using the Database Browser	10-48
Functions of the Database Browser.....	10-49
Troubleshooting	10-56

11 Enabling Transactions

Overview of Transactions	11-2
What is a Transaction?	11-2
When to use the Transaction Service	11-3
Types of Transactions.....	11-3
Components Involved in the Transaction	11-4

Configuring Your Transactional Environment	11-6
Object Transaction Service	11-7
Configuring the DAD/Resource Manager	11-8
Configuring the Distributed Transaction Coordinator	11-12
Configuring the HO Agent	11-18
Configuring the Transaction Application	11-20
Managing Your OTS Environment.....	11-28
Starting OTS Entities	11-28
Stopping OTS Entities	11-28
Deleting and Browsing Transaction Entities	11-29
 12 Failure Recovery for Oracle Application Server	
Failure Recovery Components	12-1
Failure Recovery Architecture	12-2
Failure Recovery Protocols.....	12-4
Failure Recovery of Individual Components	12-4
 A Command Line Utilities	
The owscctl utility	A-2
Version Number	A-3
version	A-3
start	A-4
stop.....	A-6
status.....	A-8
reload.....	A-11
dump	A-12
monitor	A-14
getIOR	A-16
WRB, ORB and Cartridge Processes	A-17
Files	A-19
The owstat utility	A-19
Syntax	A-19
Arguments	A-19
Environment.....	A-20

Trigger Environment Variables	A-21
Files	A-22
The oasnetconf utility	A-22
Syntax	A-23
Arguments	A-24
The oaspasswd utility	A-25
Syntax	A-26
Arguments	A-26
Restricting Access to the oaspasswd Utility	A-26
Examples	A-27
The oasmcastcfg utility	A-28
Syntax	A-28
Arguments	A-28
Examples	A-28

B Configuration and Log Files

Windows NT Files	B-1
Unix Files	B-3

Index

Preface

Audience

This guide is for system administrators who will manage, administer, and configure Oracle Application Server 4.0, HTTP listeners, and applications

The Oracle Application Server Documentation Set

This table lists the Oracle Application Server documentation set.

Title of Book	Part No.
Oracle Application Server 4.0.8 Documentation Set	A66971-03
Oracle Application Server Overview and Glossary	A60115-03
Oracle Application Server Installation Guide for Sun SPARC Solaris 2.x	A58755-03
Oracle Application Server Installation Guide for Windows NT	A58756-03
Oracle Application Server Administration Guide	A60172-03
Oracle Application Server Security Guide	A60116-03
Oracle Application Server Performance and Tuning Guide	A60120-03
Oracle Application Server Developer's Guide: PL/SQL and ODBC Applications	A66958-02
Oracle Application Server Developer's Guide: JServlet Applications	A73043-01
Oracle Application Server Developer's Guide: LiveHTML and Perl Applications	A66960-02
Oracle Application Server Developer's Guide: EJB, ECO/Java and CORBA Applications	A69966-01
Oracle Application Server Developer's Guide: C++ CORBA Applications	A70039-01
Oracle Application Server PL/SQL Web Toolkit Reference	A60123-03
Oracle Application Server PL/SQL Web Toolkit Quick Reference	A60119-03

Title of Book	Part No.
Oracle Application Server JServlet Toolkit Reference	A73045-01
Oracle Application Server JServlet Toolkit Quick Reference	A73044-01
Oracle Application Server Cartridge Management Framework	A58703-03
Oracle Application Server 4.0.8.1 Release Notes	A66106-04

Conventions

This table lists the typographical conventions used in this manual.

Convention	Example	Explanation
bold	oas.h owsctl wrbcfg www.oracle.com	Identifies file names, utilities, processes, and URLs
italics	<i>file1</i>	Identifies a variable in text; replace this place holder with a specific value or string.
angle brackets	<filename>	Identifies a variable in code; replace this place holder with a specific value or string.
courier	owsctl start wrb	Text to be entered exactly as it appears. Also used for functions.
square brackets	[-c string] [on off]	Identifies an optional item. Identifies a choice of optional items, each separated by a vertical bar (), any one option can be specified.
braces	{yes no}	Identifies a choice of mandatory items, each separated by a vertical bar ().
ellipses	n,...	Indicates that the preceding item can be repeated any number of times.

The term “Oracle Server” refers to the database server product from Oracle Corporation.

The term “**oracle**” refers to an executable or account by that name.

The term “*oracle*” refers to the owner of the Oracle software.

Technical Support Information

Oracle Global Support can be reached at the following numbers:

- In the USA: **Telephone: 1.650.506.1500**
- In Europe: **Telephone: +44 1344 860160**
- In Asia-Pacific: **Telephone: +61. 3 9246 0400**

Please prepare the following information before you call, using this page as a check-list:

- ☐ your CSI number (if applicable) or full contact details, including any special project information
- ☐ the complete release numbers of the Oracle Application Server and associated products
- ☐ the operating system name and version number
- ☐ details of error codes and numbers and descriptions. Please write these down as they occur. They are critical in helping WWCS to quickly resolve your problem.
- ☐ a full description of the issue, including:
 - **What** - What happened? For example, the command used and its result.
 - **When** - When did it happen? For example, during peak system load, or after a certain command, or after an operating system upgrade.
 - **Where** - Where did it happen? For example, on a particular system or within a certain procedure or table.
 - **Extent** - What is the extent of the problem? For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.
- ☐ Keep copies of any trace files, core dumps, and redo log files recorded at or near the time of the incident. WWCS may need these to further investigate your problem. For a list of trace and log files, see “Configuration and Log Files” in the *Administration Guide*.

For installation-related problems, please have the following additional information available:

- ☐ listings of the contents of \$ORACLE_HOME (Unix) or %ORACLE_HOME% (NT) and any staging area, if used.

-
- ❑ installation logs (**install.log**, **sql.log**, **make.log**, and **os.log**) typically stored in the **\$ORACLE_HOME/orainst** (Unix) or **%ORACLE_HOME%\orainst** (NT) directory.

Documentation Sales and Client Relations

In the United States:

- To order hardcopy documentation, call Documentation Sales: **1.800.252.0303**.
- For shipping inquiries, product exchanges, or returns, call Client Relations: **1.650.506.1500**.

In the United Kingdom:

- To order hardcopy documentation, call Oracle Direct Response: **+44 990 332200**.
- For shipping inquiries and upgrade requests, call Customer Relations: **+44 990 622300**.

Reader's Comment Form

Oracle Application Server 4.0 Administration Guide

Part No. A60172-03

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Oracle Application Server Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065

If you would like a reply, please provide your name, address, and telephone number below:

Thank you for helping us improve our documentation.

Overview

This chapter introduces the components of Oracle Application Server and summarizes the functions of Oracle Application Server Manager and Oracle Application Server Utilities.

Contents

- [Component Overview](#)
- [Using Oracle Application Server](#)
- [Oracle Application Server Welcome Page](#)
- [Oracle Application Server Manager \(OAS Manager\)](#)
- [Oracle Application Server Utilities \(OAS Utilities\)](#)

Component Overview

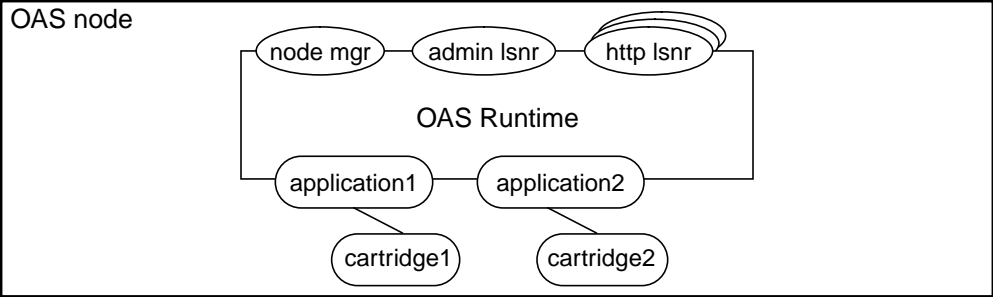
A web site is a group of networked machines (nodes) sharing the same Web Request Broker (WRB). The Web Request Broker is a Oracle Application Server runtime process that manages resources.

Each node in a web site can have the following components:

- **listeners** — receives incoming requests
- **runtime processes** — initializes necessary components for requests
- **Object Request Broker (ORB)** — manages communication between nodes
- **applications** — provides a runtime environment in which the cartridges execute
- **cartridges** — executes program logic

These parts can be located on a single machine, or can be distributed across multiple machines in a multi-node configuration.

Figure 1–1 *Components of an Oracle Application Server web site*



The HTTP listeners are the entry points into the application server. After a listener receives a request for an application, it passes the request to the runtime processes. These processes initiate any applications and their cartridge instances if necessary. Next, the application is executed which provides a runtime environment for the cartridges to run in.

For more information about the architecture of Oracle Application Server, refer to *Overview of Oracle Application Server*.

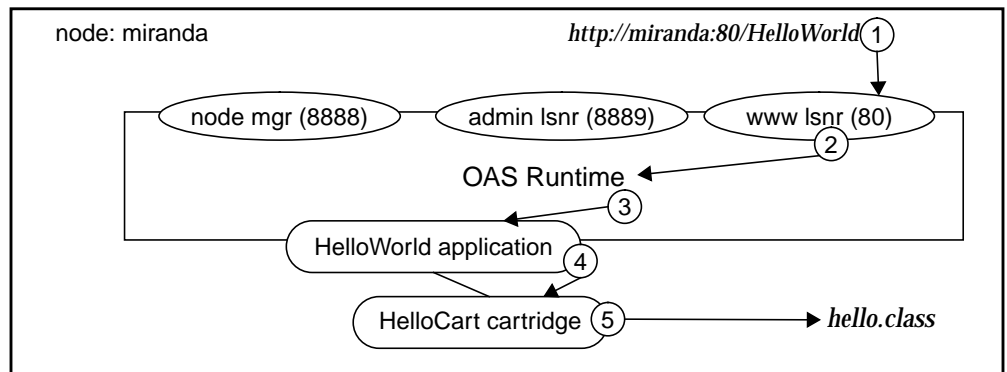
Sample Path of a Request

Figure 1–2 illustrates the path that a request to an application must follow. In this example, the web site is hosted by a machine **miranda**. It uses the default port numbers for its listeners:

Table 1–1 *Oracle Application Server listeners and default port numbers*

Listener	Port
node manager	8888
admin	8889
www	80

The site has an application named Hello World that executes the HelloCart JServlet cartridge. The program logic for the cartridge is stored in **hello.class**, a compiled Java program.

Figure 1–2 Sample path of a request

1. The client request for `http://miranda:80/HelloWorld` is received by the `www` listener on port 80.
2. The listener passes the request to the runtime processes. These processes determine that the `HelloWorld` application has a cartridge named `HelloCart`. The processes determine if there are any free instances of `HelloCart`. If there are none, an instance is created.
3. The OAS runtime starts the `HelloWorld` application, which creates a runtime environment for `HelloCart`. Since `HelloCart` is a `JServlet` cartridge, a Java virtual machine is started.
4. The `HelloCart` instance is executed by the `HelloWorld` application.
5. The `HelloCart` instance executes the program logic in `hello.class` and returns the results back through the runtime and listener to the user.

Using Oracle Application Server

To start using Oracle Application Server, you will need to perform some basic configuration. These steps will guide you through the process of creating and configuring your web site.

1. Add nodes if you want a multi-node site.

See [Chapter 2, "Managing your Site and Server"](#) for more information about adding nodes to your web site.

2. Add and configure HTTP listeners.

See [Chapter 3, "Managing and Configuring HTTP Listeners"](#) for more information about adding and configuring listeners.

3. Configure the Object Request Broker.

See [Chapter 5, "Object Request Broker Administration"](#) for more information about configuring the Object Request Broker.

4. Add and configure applications.

See [Chapter 8, "Application Administration"](#) for more information about applications.

5. Add and configure cartridges.

See [Chapter 9, "Cartridge and Component Administration"](#) for more information about cartridges.

Once your web site is running, you can further customize it by:

- configuring logging

See [Chapter 10, "Logging and Database Utilities"](#) for more information about logging.

- adding security features

See the *Security Guide* for more information about using the security features of Oracle Application Server.

- tuning performance

See the *Performance and Tuning Guide* for more information about optimizing the performance of Oracle Application Server.

Oracle Application Server Welcome Page

The Oracle Application Server welcome page is the first page that your site administrator sees when connecting to the node manager. It contains links to features of the application server. The two main configuration links from the welcome page are the OAS Manager and the OAS Utilities pages.

Figure 1–3 The Oracle Application Server Welcome Page



- The **OAS Manager** is used to configure and manage your web site. From it, an administrator can start, stop and manage the Web Request Broker (WRB) and Oracle Application Server processes. Applications and listeners can also be added, deleted or configured with the OAS Manager.
- The **OAS Utilities** are used to perform additional tasks such as installing the database packages, selecting a tuning method, and running the log analyzer.

Accessing the Welcome Page

To access the Oracle Application Server Welcome Page:

1. Launch your web browser and connect to the Welcome page at:

`http://hostname.domain:port`

where *hostname.domain* is the fully-qualified name of your machine, and *port* is the port number where your Node Manager listener runs. If you accepted the default port number during installation, the port number will be 8888.

Note: The Node Manager listener must be running to connect to the Oracle Application Server Welcome page. Refer to [Starting and Stopping the Node Manager Listener](#) on page 2-3 for more information about starting the Node Manager listener.

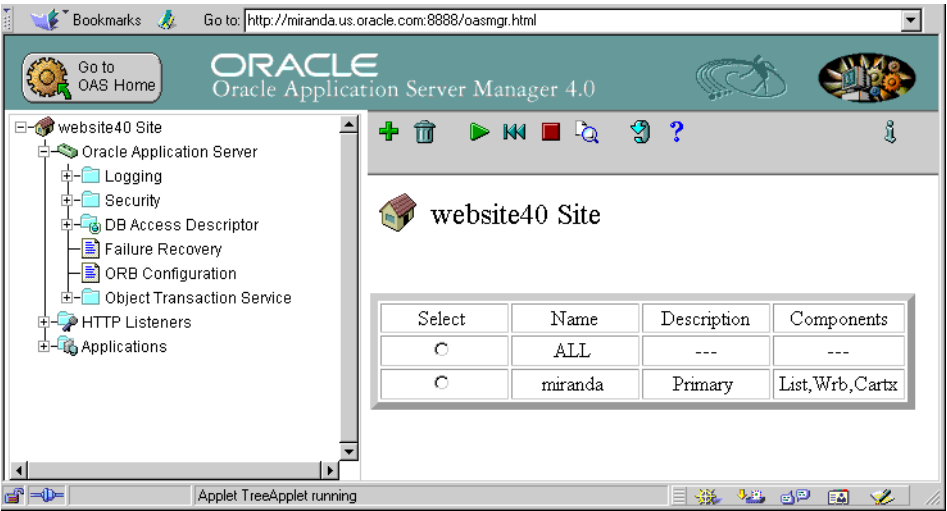
- 2. Enter the username and password for the Node Manager listener. The default username is **admin**. This option was configured during the installation procedure.

The Oracle Application Server Manager Welcome page is displayed.

Oracle Application Server Manager (OAS Manager)

Oracle Application Server Manager is a navigational tool providing access for managing your web site and Oracle Application Server. The interface contains a left frame, called the navigational tree, showing the components and features of your web site. The right frame, called the form area, displays the work area where you can view and manipulate the features of your web site and server.

Figure 1–4 The Oracle Application Server Manager



By expanding the navigation tree in the left frame, the sub-components of a site are revealed. The top level of this tree is the web site level. When you click on a leaf in









the navigation tree, the form corresponding to that component is opened in the right frame. The component form lets you configure and manage the selected component.

Node Manager Operations and Icons

In the node manager, there are several operations you can perform from the form area. [Table 1-2](#) lists the basic operations that are available with most forms and their icons.

Depending on which branch of the navigation tree is selected, the operations will affect different components of Oracle Application Server. If the Site Manager is selected, the operations will affect nodes. If Oracle Application Server is selected, the operations will affect processes. Similarly, if either HTTP Listeners or Applications is selected, that component will be affected.

Table 1-2 *Form operations*

Operation	Icon	Comments
Add		Allows you to add a new component.
Delete		Removes the selected component.
Stop		Stops the selected components.
Reload		Reloads (stops and then starts) the configuration information for the selected component.
Start		Starts the selected component.
Monitor		Displays information about the selected component.
Update Page		Updates the current form.
Help		Provides help for the current form.

Reloading and Refreshing the Oracle Application Server

With Oracle Application Server you can reload components and refresh the interface.

- Reloading the application server involves stopping and restarting the application server at the server level. This can be done in one step by using the reload button (⏮) at the web site level.

Pressing the Oracle Application Server reload button (⏮) at the Oracle Application Server Oracle Application Server level reloads the server processes.

- Refreshing the Oracle Application Server navigational tree and forms at the browser level by pressing Shift + Reload in your browser's toolbar.

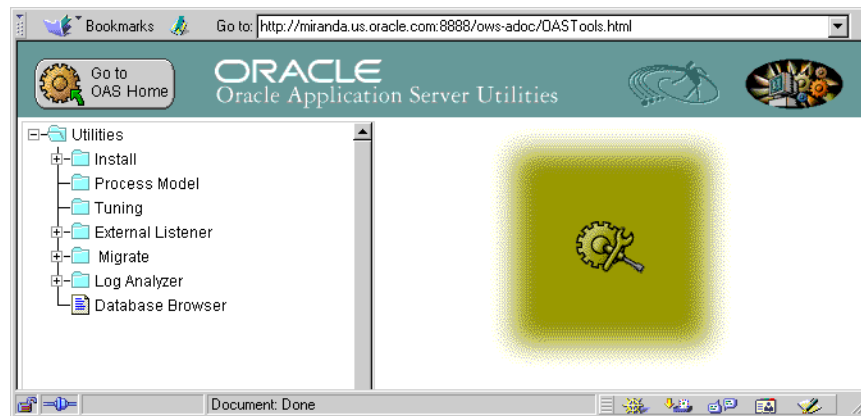
After adding or deleting any component, you must refresh the navigational tree to view your changes.

In order for configuration changes to take effect, click the Reload button at both the web site level and the HTTP Listener level. Simply refreshing the navigational tree does not automatically reload the server processes.

Oracle Application Server Utilities (OAS Utilities)

The OAS Utilities allow you to perform a number of important functions that effect your web site, application server, and database.

Figure 1–5 The Oracle Application Server Utilities



Install

Expanding the Install folder displays installation forms for the Oracle Application Server logging utilities. These forms allow you to install and configure items that were not installed initially with Oracle Application Server.

See [Chapter 10, "Logging and Database Utilities"](#) for more information about installing Oracle Application Server logging utilities.

Tuning

The Tuning form allows you to select a method of load balancing: Priority based or Min/Max based. Priority based tuning manages and allocates your system resources automatically based on the priority level you set for your applications and cartridges. Min/Max load tuning is a manual task requiring setting tuning parameters at the application and cartridge levels.

See the *Performance and Tuning Guide* for more information about tuning models.

External Listener

The External Listener forms allow the administrator to register a third-party listener with Oracle Application Server.

See the *Installation Guide* for more information about external listeners.

Migrate

Use the Migrate forms to upgrade your installation from previous Oracle Application Server versions. With these forms, you can upgrade the WRB, listeners, and cartridges.

See the *Installation Guide* for more information about migrating from previous versions of Oracle Application Server.

Log Analyzer

Use the Log Analyzer forms to upload log files into the database and run reports.

See [Chapter 10, "Logging and Database Utilities"](#) for more information about the log analyzer.

Database Browser

The Database Browser forms allow you to view information about your database. With it, you can access information about a database's tables, views, stored PL/SQL procedures, users, sessions, rollback segments, files and tablespace statistics.

See [Chapter 10, "Logging and Database Utilities"](#) for more information about the database browser.

Managing your Site and Server

This chapter describes the basic operations you can perform to the internal components of your web site and Oracle Application Server.

Contents

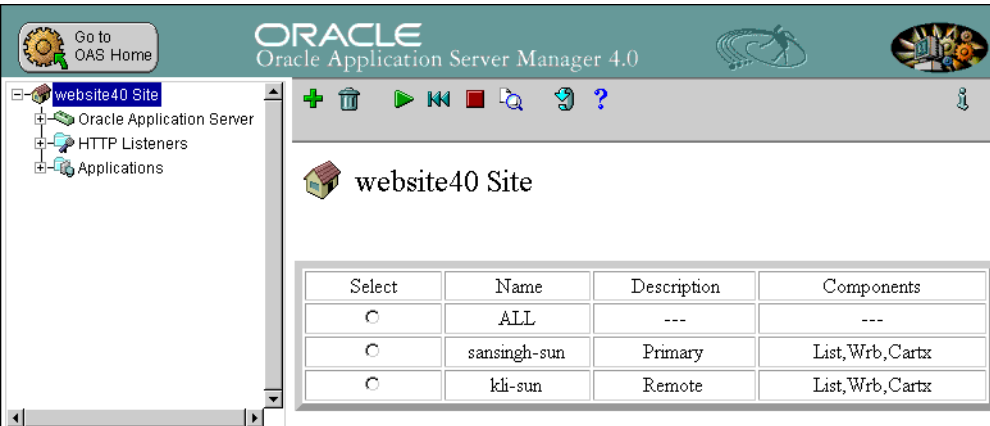
- [Understanding your Web Site and Oracle Application Server](#)
- [Understanding the OAS Manager Listeners](#)
- [Accessing the Oracle Application Server Manager](#)
- [Configuring Nodes on your Web Site](#)
- [Configuring Oracle Application Server Processes](#)
- [Managing Oracle Application Server Processes](#)
- [Monitoring your Oracle Application Server](#)

Understanding your Web Site and Oracle Application Server

The Oracle Application Server Manager presents Oracle Application Server components in a tree hierarchy. As you expand the tree, you gain access to configuration information for specific components.

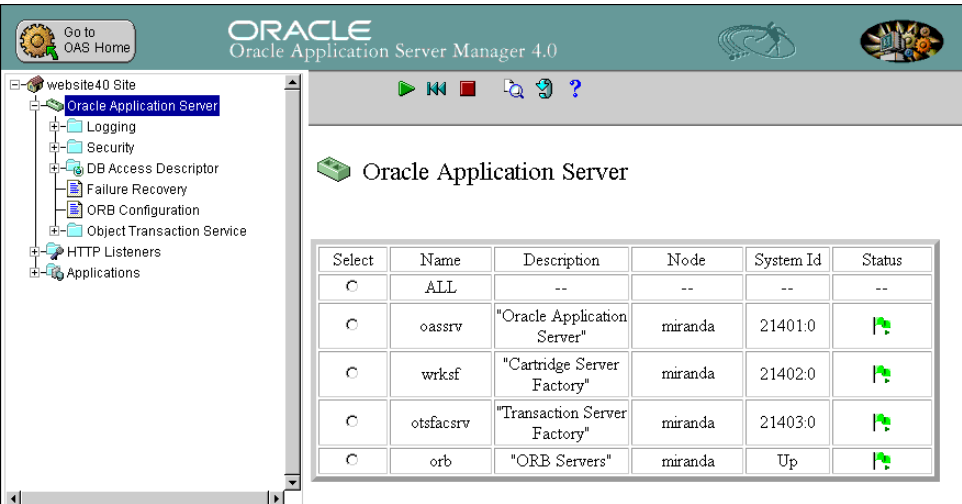
When you first load the application server, you can access the Site Manager form by selecting the web site name which defaults to “website40 site”. From the Site Manager form, you can access all of the nodes and processes in the system. You can also add or delete nodes from your web site or start, stop and monitor the site processes.

Figure 2–1 Site Manager form



By expanding the web site folder, you access the components of the web site. Now you can access the Oracle Application Server form. This form allows you to monitor, manage, and configure individual server processes.

Figure 2–2 Oracle Application Server form



For more information about configuring or managing Oracle Application Server processes, please refer to [Managing Oracle Application Server Processes](#) on page 2-9.

Understanding the OAS Manager Listeners

The Oracle Application Server Manager depends on two manager listeners:

- [Node Manager Listener](#)
- [Administration Utility Listener](#)

The Administration Utility listener resides on the primary node of your web site. A Node Manager listener resides on the primary node and on each remote node.

Note: Oracle Application Server uses a third listener, the HTTP listener. This is not a manager listener and will be described in [Chapter 3, "Managing and Configuring HTTP Listeners"](#).

Node Manager Listener

A Node Manager listener is present on each node in your web site. The graphical interface uses this listener to accept configuration changes. Because this listener handles requests to the interface, it must be managed from a command prompt.

Checking the Status of the Node Manager Listener

To check the status of your Node Manager listener, at the command line, type:

```
% owsctl status -nodemgr
```

Starting and Stopping the Node Manager Listener

If you need to start a Node Manager listener, type:

```
% owsctl start -nodemgr
```

To stop a Node Manager listener, type:

```
% owsctl stop -nodemgr
```

To manage and configure all other listeners, use the Oracle Application Server Manager, as described below and in [Chapter 3, "Managing and Configuring HTTP Listeners"](#).

More information on `owsctl` and other command line utilities can be found in [Appendix A, "Command Line Utilities"](#).

Administration Utility Listener

The Administration Utility listener allows you to use the Oracle Application Server Manager to run the Log Analyzer and Database Browser. It also allows you to run Oracle Application Server samples and demos. The Administration Utility listener is found only on the primary node; it is not present on remote nodes.

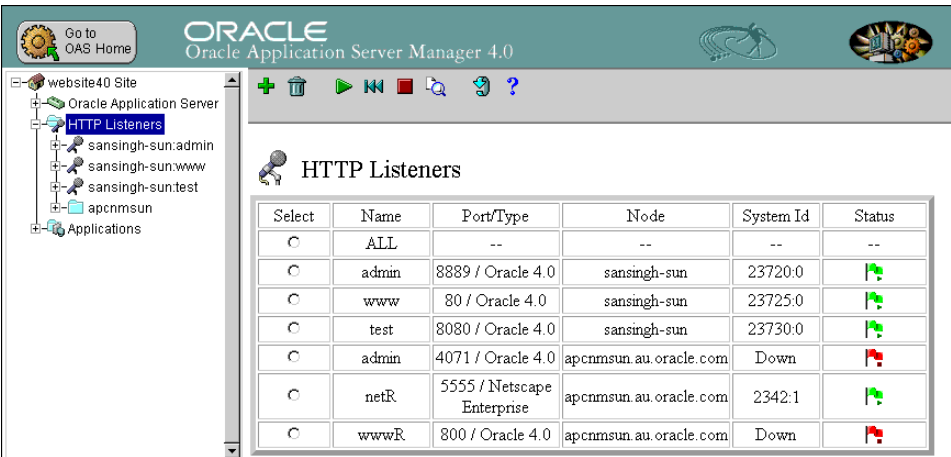
Unlike the Node Manager listener, the Administration Utility listener depends on the Oracle Application Server processes to function. Start the application server processes before starting the Administration Utility listener.

Checking the Status of the Administration Utility Listener

To check the status of the Administration Utility listener:

1. Connect to the OAS Manager.
2. Expand the site name navigational tree and the HTTP Listeners folder.

Figure 2–3 HTTP Listeners form




3. Click the *hostname:admin* icon. This opens the control form for the Administration Utility listener.

The HTTP Listeners form for the Administration Utility listener is displayed.

The status of the listener is displayed in the Status column. A green flag indicates that the listener is up; a red flag means that it is down.

Starting the Administration Utility Listener

To start the Administration Utility listener:

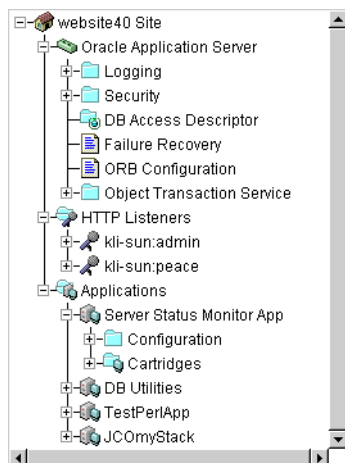
1. Expand the web site folder in the Node Manager to reveal HTTP Listeners.
2. Select *hostname:admin*. This will load the Administration Utility listener form in the right frame.
3. Select the radio button next to admin, and click the Start button () to start the listener.

Accessing the Oracle Application Server Manager

The Site Manager form ([Figure 2-5](#)) allows you to use several management operations to manage your web site. To access the Site Manager:

1. Access the Oracle Application Server Manager Welcome page. See "[Accessing the Welcome Page](#)" on page 1-5 for instructions.
2. Click on OAS Manager to use the Oracle Application Server Manager. Click on OAS Utilities to use the Oracle Application Server utilities.
3. Click on OAS Manager and expand the web site level. The OAS Manager navigational tree is shown in [Figure 2-4](#).

Note: You can reload the navigational tree after a configuration change by holding down the Shift key and pressing the browser Reload button. This forces the applet to reload and display the configuration information.

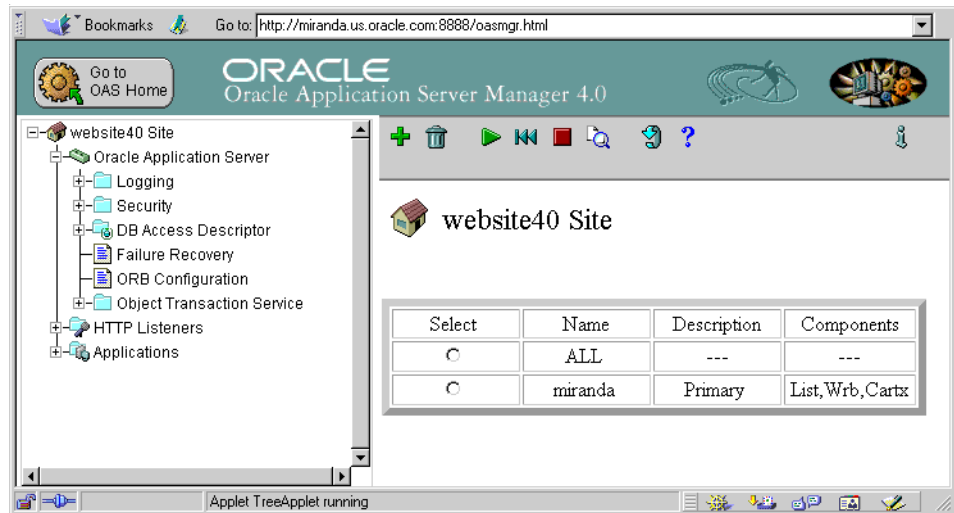
Figure 2–4 OAS Manager navigational tree

Configuring Nodes on your Web Site

A web site is a group of nodes sharing the same Web Request Broker (WRB). A node is a host computer with a unique domain name. You can have two types of nodes: primary and remote. The primary node contains the WRB, listeners, and optionally, cartridges. The remote node can contain a combination of listeners, cartridges and a subset of the WRB. The Site Manager form provides several operations you can apply to the nodes belonging to a site.

For large workloads, you can improve performance by setting up your site with multiple nodes. For example, you could be running listeners and WRB processes on the primary node, and running cartridge servers on remote nodes. You can optimize performance by dividing responsibilities among nodes according to the workloads you anticipate.

For more information about node configurations and performance issues, refer to the *Performance and Tuning Guide*.

Figure 2–5 Site Manager form

The Site Manager form contains a table, which describes all the nodes belonging to a site:

Operation	Comments
Select	Allows you to select a particular node so that you can use the operations mentioned above.
Name	Contains the node names of all the registered nodes for this site.
Description	Provides more information about the node type: primary or remote.
Components	Lists all components (WRB processes, Object Request Broker (ORB), etc.) that are running on the node.

Adding a New Node

You can add remote nodes to your site if you chose a multi-node installation at installation time. The node you will be adding must already be installed as a remote node. See the *Installation Guide* for information about multi-node installations.

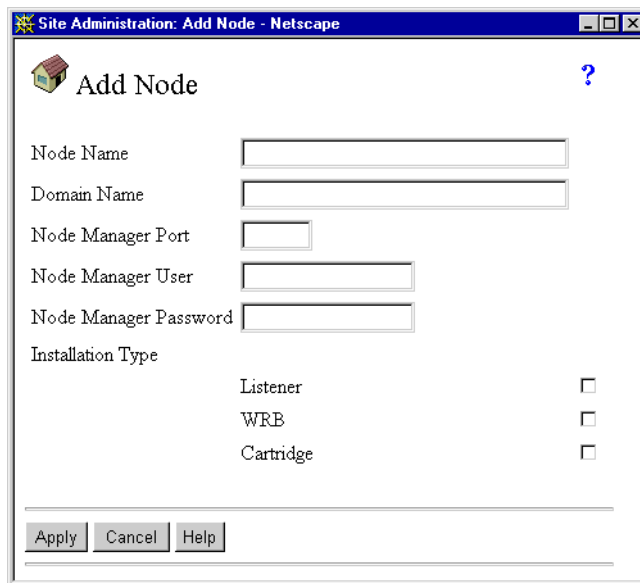
To add a new node:

1. Click on web site name in the Oracle Application Server Manager navigation tree.

2. Click on the Add button (+) to bring up the Add Node form.

The Add Node form (Figure 2–6) allows you to add a new node to your web site.

Figure 2–6 Add Node



The screenshot shows a Netscape browser window titled "Site Administration: Add Node - Netscape". The main content area is titled "Add Node" with a house icon and a question mark. It contains several input fields: "Node Name", "Domain Name", "Node Manager Port", "Node Manager User", and "Node Manager Password". Below these is a section for "Installation Type" with three radio button options: "Listener", "WRB", and "Cartridge". At the bottom are three buttons: "Apply", "Cancel", and "Help".

Node Name The machine name, no longer than 30 characters, that uniquely identifies the machine name. For example, to add a node with hostname `miranda.us.oracle.com`, the node name would be *miranda*.

Note: When installing a node from a different domain, you must enter the full hostname as the node name.

Domain Name Enter the domain name by which the machine identifies itself when serving requests. This must be a valid name recognized by DNS. From the previous example, the domain name would be *us.oracle.com*.

Node Manager Port A TCP/IP port identifying the port number of the Node Manager listener. This was set when the remote node was installed.

Node Manager User Enter the user identity to access the remote node's Node Manager listener.

Node Manager Password Enter the password for the Node Manager User.

Installation Type Choose the components that were installed with the remote node. See the *Installation Guide* for information about installing remote nodes.

- **Listener** — Allows you to access the HTTP listeners from the remote node at your primary node. If the HTTP listener component is installed, the listener functions (requests and responses) will be performed on the remote node.
- **WRB** — Allows you to run any WRB process or component from the remote node on the primary node. This can be useful for load balancing the log analyzer or RM Proxy service.
- **Cartridge** — Allows you to run cartridges on the remote node. This allows your remote node to provide resources for applications.

More information about these components can be found in the *Performance and Tuning Guide*.

Configuring Oracle Application Server Processes

The WRB is the central component of Oracle Application Server. The Oracle Application server is oassrv. This process includes the logger, resource manager and monitoring functions. Refer to *Oracle Application Server Overview and Glossary* for more information about the specific Oracle Application Server functions.

Managing Oracle Application Server Processes

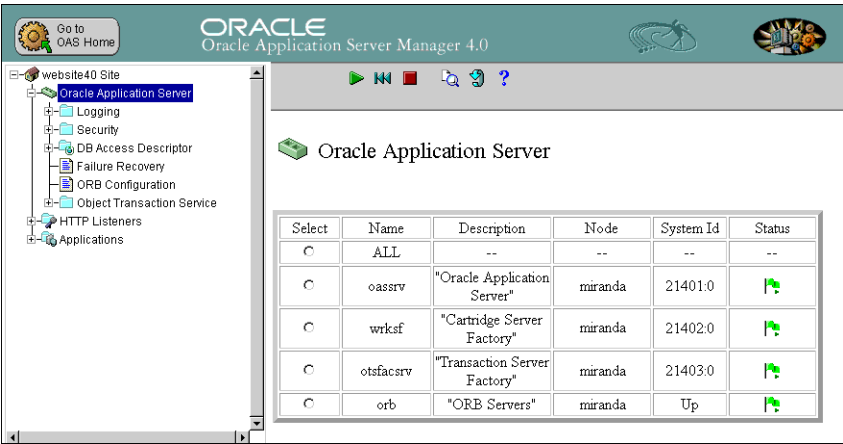
Although a web site shares a single Web Request Broker (WRB), the processes may be distributed among the nodes in the site. See the *Performance and Tuning Guide* for more information about distributing processes.

The Oracle Application Server form ([Figure 2-7](#)) allows you to use several management operations to manage your WRB processes. You can stop, reload, start, or monitor any WRB process.

1. Expand the web site folder by clicking on the .

- 2. Click on the Oracle Application Server form name to bring up the Oracle Application Server form.

Figure 2-7 Oracle Application Server form




The Oracle Application Server form contains a table, which describes all the WRB processes belonging to a site:

Operation	Comments
Select	Allows you to select a particular process so that you can use the operations mentioned above.
Name	Contains the executable name of all the processes running on this site. For example, wrblog.
Description	Provides more descriptive information about the process.
Node	Specifies the different machine names. For example, miranda.
System Id	Indicates the process id and instance. For example, 2342:1 where 2342 is the process id and 1 is the instance of the process. If a process is down, it does not display a system ID.
Status	<p>Indicates whether the process is up or down. A red flag means the process is down; a green flag means the process is up.</p> <p>If wrbmon is down, the status of all the other processes are unknown and shows them as down. If you think your processes are up, try to start wrbmon first. The oassrv process has to be running since wrbmon is part of oassrv.</p>

Starting the Oracle Application Server Processes

To start the Oracle Application Server processes:

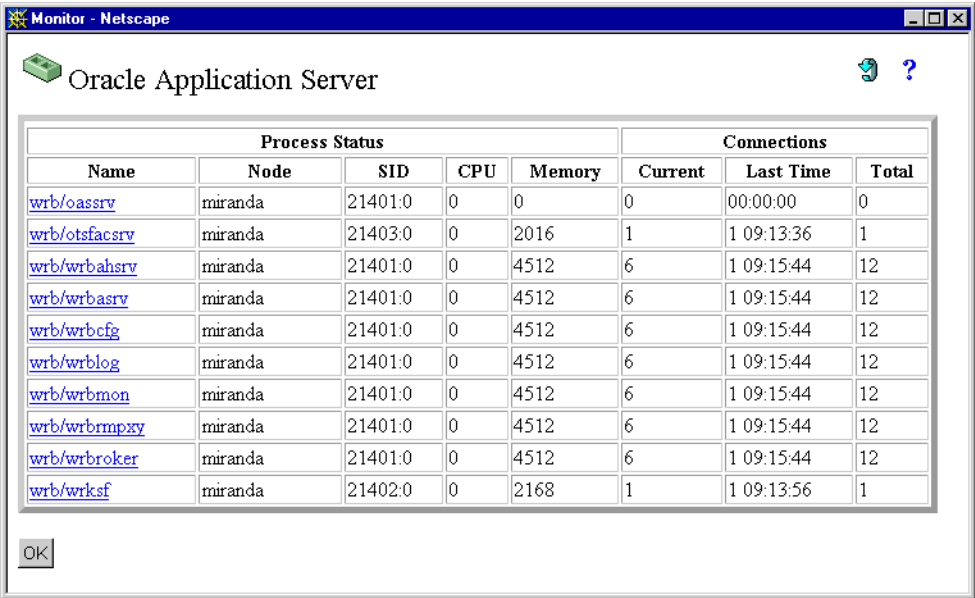
1. Expand the web site folder in the Node Manager to reveal Oracle Application Server.
2. Select Oracle Application Server from the navigation tree.
3. In the Oracle Application Server form, select the radio button next to ALL.
4. Click the Start button () to start all of the Oracle Application Server processes.

Monitoring your Oracle Application Server

The Oracle Application Server Monitor form ([Figure 2-8](#)) displays the results for the process you selected to monitor. See "[Accessing the Oracle Application Server Monitoring Window](#)" on page 2-13 for instructions on bringing up this window.

Note: All the monitoring statistics are a rough indication of the state of the server. They are not precise. For example, the total number of requests completed include other internal calls of the Oracle Application Server.

Figure 2–8 Oracle Application Server monitoring form



The following table describes the entries which display process status information.

Column	Description
Name	The name of the process.
Node	The name of the machine.
SID	The process ID and (after the colon) the instance ID.
CPU	The amount of time used by the process in 1/100th of a second.
Memory	The process heap memory used in kilobytes (KB).


Note: if the **wrbmon** service is not available, the Oracle Application Server Manager may display incorrect or outdated status information. This happens when the **oassrv** process is not running. Start the **oassrv** process from the Oracle Application Server form ([Figure 2–7](#)).

The following table describes the entries which display connection information.

Column	Description
Current	The number of connections to the process at that moment.
Last Time	The time of the last connection to the process in the format [[(days)hour:minutes:seconds]. This interval is measured since the process started.
Total	The total number of current connections.

Accessing the Oracle Application Server Monitoring Window

From the Oracle Application Server Manager, complete the following steps to bring up the monitoring window.

1. In the navigation tree, select Oracle Application Server. (You may need to expand the web site folder to see this option.)
2. When the list of processes appears in the right frame, choose the process you want to monitor by selecting the radio button next to its name. If you want to monitor all processes, select the radio button next to ALL.
3. Press the Monitor button  in the tool bar.

The Oracle Application Server monitoring window will appear.

Managing and Configuring HTTP Listeners

This chapter describes the basic operations to manage, monitor, administer and configure your HTTP listeners. An HTTP listener listens for HTTP requests and handles only requests for static documents and CGI scripts. All other requests are forwarded to the Dispatcher, which handles the server-side applications. You can run several HTTP listeners at once, each accepting requests from a different port.

For more information about HTTP listeners, refer to [Chapter 1, "Overview"](#).

Contents

- [Multiple Addresses and Ports](#)
- [Administering Your HTTP Listeners](#)
- [Monitoring Your HTTP Listeners](#)
- [Adding a New Listener](#)
- [Configuring a Listener](#)
- [Using Third-Party Listeners](#)

Multiple Addresses and Ports

When you create a new listener, it responds only to requests directed at a single IP address and port. This listener can be configured to respond to requests directed at any number of addresses and ports. Different response behaviors for each of these address/port combinations can be configured as well. A single listener that can handle requests directed at more than one address/port combination is called a multiport listener. The address/port combination is called a virtual host.

The advantages of multiport listeners is that Oracle Application Server can treat requests differently depending upon which virtual host the request is directed at in terms of the following:

- Security applied to the request
- The file system which provides information for the request
- The log information collected about the request

For information about configuring virtual hosts, see [Configuring Network Parameters](#) on page 3-8.

Duplicate Port Numbers and IP Addresses

You can specify more than one IP address per port and more than one port per IP address. However, you cannot specify the same IP address and port number combination more than once on the same host. For example, you can use a configuration with two ports on the same host:

Address	Port	Security	Host Name	Base Dir	Log Info	Authen
188.4.219.42	8443	NORM	www.xyz.com	/scripts/	/tmp/mylogs/	NONE
188.4.219.42	2021	NORM	www.xyz.xom	/sample/a/	/tmp/mylogs/	NONE

or one with the same port on different hosts:

188.4.219.42	8443	NORM	www.xyz.com	/scripts/	/tmp/mylogs/	NONE
188.4.219.42	8443	NORM	foo.xyz.com	/sample/a/	/tmp/mylogs/	NONE

but cannot use one with more than one use of the same port and host information:

188.4.219.42	8443	NORM	www.xyz.com	/scripts/	/tmp/mylogs/	NONE
188.4.219.42	8443	NORM	www.xyz.com	/sample/a/	/tmp/mylogs/	NONE

How Oracle Application Server Manages Multiport Listeners

Oracle Application Server manages all virtual hosts for a listener as a suite so that any operation it applies to one virtual host is applied equally to all other virtual hosts. For example, when the listener is stopped, all of that listener’s virtual hosts stop. Similarly, when the listener starts, it starts listening on all of its virtual hosts.

In addition, Oracle Application Server considers an error that occurs with any virtual host as an error for the listener. The server reacts accordingly in this case. For example, if one of the listener’s virtual hosts is inaccessible when you try to start the listener, the listener will fail to start, even though it may be able to access the

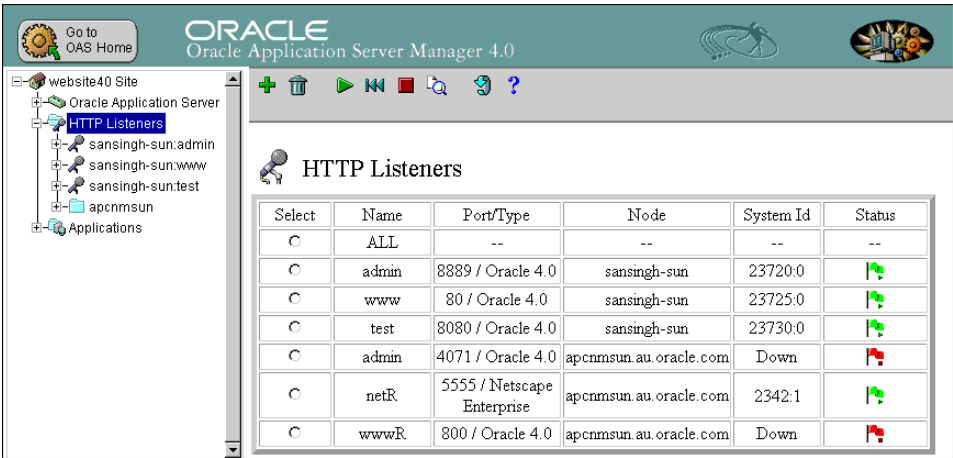
other virtual hosts. In this case, the server does not start until it reads a new configuration file with a correct set of virtual hosts on which it can listen.

Administering Your HTTP Listeners

The HTTP Listeners form (Figure 3–1) allows you to manage your HTTP listeners. You can create, start, stop, or delete an HTTP Listener process. To access the HTTP Listener form:

- 1. Choose OAS Manager from the Welcome Page.
- 2. Expand the website40 Site form.
- 3. Click on the HTTP Listeners form name to bring up the HTTP Listener form.

Figure 3–1 HTTP Listeners form



The HTTP Listeners form contains a table, which describes all the listeners belonging to a site:

Operation	Comments
Select	Allows you to select a particular listener so that you can use the operations mentioned above.
Name	Contains the listener name of all registered listeners in this site.

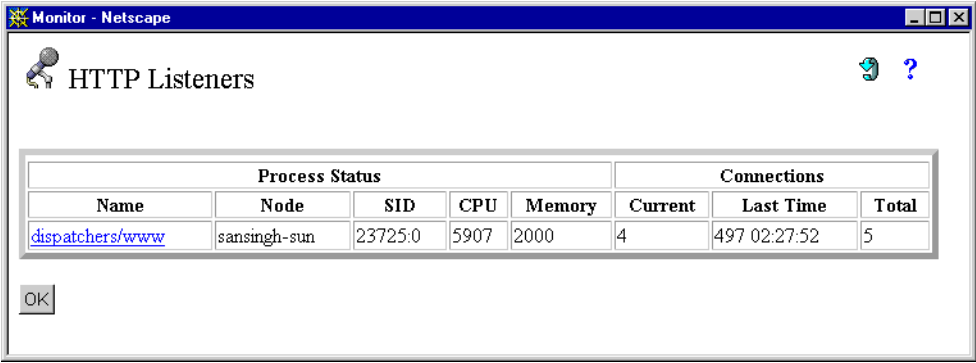
Operation	Comments
Port/Type	Provides the port number for the listener and the listener type. For example, 5555/Netscape Enterprise indicates that port 5555 uses a Netscape Enterprise listener.
Node	Specifies the different machine names. For example, apcnmsun.au.oracle.com.
System Id	Indicates the process id and instance. For example, 2342:1 where 2342 is the process id and 1 is the instance of the process. If a process is down, it does not display a system ID.
Status	Indicates whether the listener is up or down. A red flag means the process is down; a green flag means the process is up.

Monitoring Your HTTP Listeners

The HTTP Listeners Monitoring window (Figure 3-2) displays the results for the listeners you selected to monitor. See "Accessing the HTTP Listeners Monitoring Window" on page 3-5 for instructions on bringing up this window.

Note: All the monitoring statistics are a rough indication of the state of the server. They are not precise. For example, the total number of requests completed include other internal calls of the Oracle Application Server.

Figure 3-2 HTTP Listeners Monitoring window



Process Status					Connections		
Name	Node	SID	CPU	Memory	Current	Last Time	Total
dispatchers/www	sansingh-sun	23725:0	5907	2000	4	497 02:27:52	5

OK

The following table describes the entries which display process status information.


Column	Description
Name	The name of the listener.
Node	The name of the machine that the listener is on.
SID	The process ID and (after the colon) the instance ID.
CPU	The amount of time used by the listener in 1/100th of a second.
Memory	The process heap memory used in kilobytes (KB).

The following table describes the entries which display connection information.


Column	Description
Current	The number of connections to the listener at that moment.
Last Time	The time of the last connection to the listener in the format [(days)hour:minutes:seconds]. This interval is measured since the listener started.
Total	The total number of current connections to the listener.

Accessing the HTTP Listeners Monitoring Window

From the Oracle Application Server Manager, complete the following steps to bring up the monitoring window.

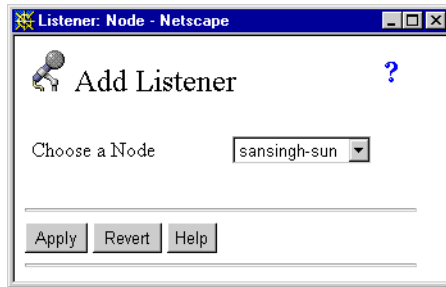
1. In the navigation tree, select HTTP Listeners. (You may need to expand the web site folder to see this option.)
2. When the list of listeners appears in the right frame, choose the listener you want to monitor by selecting the radio button next to its name. If you want to monitor all listeners, select the radio button next to ALL.
3. Press the Monitor button  in the tool bar.
4. The HTTP listeners monitoring window will appear.

Adding a New Listener

To add a new listener to your web site, click on the add button  from the HTTP Listeners form (Figure 3-1).

If you are in a multi-node configuration, you will be asked to select a node on which you want to create a new HTTP listener (Figure 3-3).

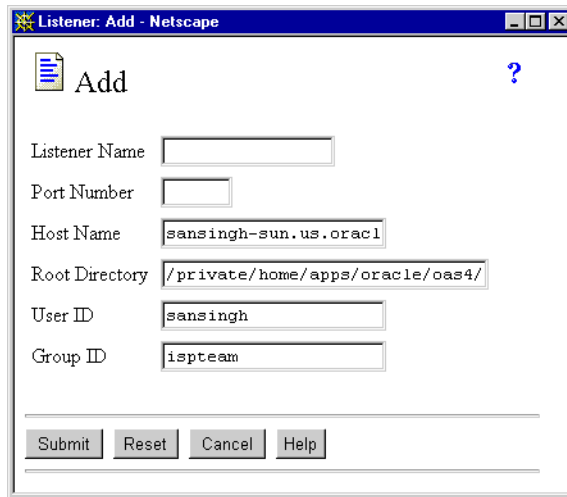
Figure 3–3 Add Listener form — choose a node



The screenshot shows a Netscape browser window titled 'Listener: Node - Netscape'. The main heading is 'Add Listener' with a question mark icon. Below the heading is a label 'Choose a Node' followed by a dropdown menu showing 'sansingh-sun'. At the bottom are three buttons: 'Apply', 'Revert', and 'Help'.

Select the machine name for the HTTP listener. After selecting the node and pressing the Apply button, the Add form (Figure 3–4) appears allowing you to create the new listener.

Figure 3–4 Add Listener form — configure the listener



The screenshot shows a Netscape browser window titled 'Listener: Add - Netscape'. The main heading is 'Add' with a question mark icon. Below the heading are several input fields: 'Listener Name' (empty), 'Port Number' (empty), 'Host Name' (filled with 'sansingh-sun.us.orac1'), 'Root Directory' (filled with '/private/home/apps/oracle/oas4/'), 'User ID' (filled with 'sansingh'), and 'Group ID' (filled with 'ispteam'). At the bottom are four buttons: 'Submit', 'Reset', 'Cancel', and 'Help'.

Listener Name An alphanumeric string of your choice no longer than six characters that uniquely identifies the HTTP listener.

Port Number A TCP/IP port on which the HTTP listener can accept connections. You can choose any port number from 1 to 65535 that is not assigned to another listener or program. To access ports 1 through 1023, the HTTP listener process must start up with root privileges (see the User ID and Group ID fields.)

Host Name The name the HTTP listener will use to identify itself. You should use your Oracle Application Server machine's primary host name such as **miranda.us.oracle.com**, or an alias such as **www.olab.com** by which you can identify your HTTP listener on the Internet. See your operating system documentation to learn how to create a hostname alias.

When you connect to this listener, you must specify the host name exactly as you enter it here. For example, if you specify your host name as **miranda.us.oracle.com** listening on port 5555, you type the following URL to access the site:


```
http://miranda.us.oracle.com:5555
```

Root Directory The file-system directory that serves as the root of the HTTP listener's virtual file system. This is the directory in the local file system to which the URL `http://hostname/` will refer. Make sure to use a "/" at the end. For example, **/private/oracle/ows/4.0/doc/**.

User ID (Unix only) The user identity that the HTTP listener assumes after it completes its configuration. You can specify either a username or numeric user ID in this field. To denote the root ID, you must specify "root" rather than "0". See ["Configuring User/Group Parameters"](#) on page 3-15 for more information about this parameter.

Group ID (Unix only) The group identity that the HTTP listener assumes after it completes its configuration. You can specify either a group name or a numeric group ID in this field. See ["Configuring User/Group Parameters"](#) on page 3-15 for more information about this parameter.

Starting the Listener

After successfully adding a new listener, you should start it. From the HTTP Listeners form ([Figure 3-1](#)), select the new listener and press the Start button (.

Configuring a Listener

After adding a listener, you can configure several parameters. These parameters allow you to control how the listener uses the network, defines a virtual file system by mapping specific virtual path names used in URLs to the file system path names of local directories, and configures various defaults for the listener. To access the configuration forms:

1. Expand the website40 Site folder by clicking on the .

- 2. Expand the HTTP Listeners folder.
- 3. Expand the <name of listener> folder.

At the listener level you have the following options.

- [Configuring Network Parameters](#)
- [Configuring Server Parameters](#)
- [Configuring User/Group Parameters](#)
- [Configuring Directory Mappings](#)
- [Configuring Language Extensions](#)
- [Configuring MIME Types](#)
- [Configuring Encoding Extensions](#)
- [SSL CA Roots](#)
- [Security](#)

Configuring Network Parameters

The Network form (Figure 3–5) allows you to specify the network identity of the host on which the HTTP listener runs, and controls how the HTTP listener uses the network. This option also allows you to configure your listener to accept requests directed at more than one virtual host.

Figure 3–5 Network form

Go to OAS Home

ORACLE
Oracle Application Server Manager 4.0

Network

Maximum # of Connections: 500

DNS Resolution: ALWAYS

URL of Redirection Server:

Listener PID file: /private/home/apps/or

Address	Port	Security	Host Name	Base Directory	Log Info Directory	Authentication	Certificate Label
ANY	8889	NORM	samsingh-sun.us.	/	/private/home/apps/oracle/	NONE	
ANY	8889	NORM	samsingh-sun	/	/private/home/apps/oracle/	NONE	
		NORM				NONE	
		NORM				NONE	

Apply Revert Help

In order for configuration changes to take effect, click the Reload button at both the Oracle Application Server level and the HTTP Listener level. Simply refreshing the navigational tree does not automatically reload the server processes.

Maximum # of Connections The maximum number of HTTP listener connections that may be active at one time. Any requests received by the listener after reaching its maximum number of connections will not be accepted by the listener. You should allow as many connections as your machine can handle simultaneously without severely impairing performance. [Table 3–1](#) shows the default and maximum values you can enter into this field.

Table 3–1 Maximum HTTP listener connections

Platform	Default parameter setting	Maximum parameter setting
Windows NT	600	2000
Unix	500	700

DNS Resolution Specifies when to translate IP addresses into DNS (Domain Name Service) hostnames. Possible values are:

- ALWAYS — Always translate IP addresses into DNS hostnames.
- LAZY — Translate IP addresses into DNS hostnames only when needed by an Application Development Interface (ADI) application or security module.
- LAZY_WITH_CGI — Translate IP addresses into DNS hostnames only when needed by CGI programs.
- NEVER — Never use DNS hostnames (default).

Setting DNS resolution to ALWAYS slows your HTTP listener's performance somewhat because the DNS server usually runs on a different machine from Oracle Application Server, making each transaction subject to a time-consuming DNS query.

On the other hand, if you use domain-based restriction for access control, do not set DNS resolution to NEVER. In this case, setting DNS Resolution to LAZY_WITH_CGI is often a good compromise.

URL of Redirection Server The URL of a HTTP listener process to which requests can be redirected when the selected listener's maximum number of connections has been reached. The URL must be of the form:

`http://HostName:PortNumber`

The redirection server must have all the capabilities of the selected HTTP listener so that it can fulfill the same requests. If you are consistently receiving the maximum number of connections on a certain HTTP listener process, you might want to create another listener by copying the original listener configuration and use the new listener as a redirection server for the original.

See the *Performance and Tuning Guide* for more information about URL redirection.

Note: The redirection server must have all the capabilities of the selected HTTP listener so that it can fulfill the same requests. If you are consistently receiving the maximum number of connections on a certain HTTP listener process, you might want to create another listener by copying the original listener configuration and use the new listener as a redirection server for the original.

Listener PID file This parameter is valid only for UNIX.

Specifies a file that contains the Process ID PID of the HTTP listener process.

Note: You must use the HTTP Listeners form to stop a HTTP listener before changing its Listener PID file. If you do not do this, the HTTP Listeners form loses track of the HTTP listener process.

Each HTTP listener is capable of accepting connections on multiple TCP/IP ports. For more information see "[Multiple Addresses and Ports](#) on page 3-1". Each entry in this section specifies the network identity the HTTP listener assumes for connections on a particular port. Each entry has the following fields:

Address The IP address associated with the specified port. On machines with more than one IP address, you may assign a different IP address to each port. A value of `zero` or `ANY` means the HTTP listener can accept connections from the specified port on any of the machine's available addresses. (See your operating system documentation if you want to learn how to configure multiple IP addresses on your machine.)

Port A TCP/IP port on which the HTTP listener can accept connections. You can choose any number from 1 to 65535. To use ports 1 through 1023, you must give the HTTP listener process root privileges.

Security Specifies the security module to handle incoming requests through the address and port combination. Possible values are:

- **NORM** — Does not support Secure-Sockets Layer (SSL) on a port.
- **SSL** — Clients using SSL 2.0 and/or SSL 3.0 are supported. Listener will understand initial SSLV2 Hello messages, but will try to negotiate up to SSL 3.0. If the client does not support SSL 3.0, then further communication will proceed using SSL 2.0.
- **SSL_V2** — Communication is possible only using SSL 2.0. No communication takes place if SSL2 is not checked in your browser configuration.
- **SSL_V3** — Communication is possible only using SSL 3.0. No communication takes place if SSL3 is not checked in your browser configuration.
- **SSL_V3_V2H** — Forces an SSL 3.0 connection, but allow a meaningful error when SSL 2.0-only clients attempt communication. Listener will understand initial SSL2.0 Hello messages, but will try to negotiate up to SSL 3.0. If the client does not support SSL3.0, then further communication will fail.

Host Name The name by which the server identifies itself on this virtual host. You can use your machine's primary hostname, or an alias such as **www.olab.com**, by which you want to identify your HTTP listener on the Internet. Refer to your operating system documentation to learn how to define a hostname alias. This must be a valid name recognized by DNS. You can specify more than one hostname (domain) and port combination (see [Duplicate Port Numbers and IP Addresses](#) on page 3-2).

Consider the following if you are configuring virtual hosts:

- Because each domain must be specified in the Host Name field, if you want to allow users to enter a partial domain, you must explicitly define the partial domain.
- Multiple domains can map to the same IP address through DNS. Oracle Application Server checks the host request-header field to determine which virtual host, or domain name, is requested.
- Each domain can supply content from its own base directory.

Base Directory The directory name to which URL-encoded path names addressed to this port are to be appended. For example, if the base directory is `/public_html`, the URL `http://www.olab.com/file` is converted to `http://www.olab.com/public_html/file`.

Note: To use the User Directory feature under the Server form, the Base Directory must be set to “/” (root). See the following section, [Configuring Server Parameters](#) for more information.

Log Info Directory The log information directory containing log information files that the listener uses to write informative messages for this port.

Authentication Determines if Oracle Application Server authenticates a client over an SSL port. Possible values are:

- NONE — Does not authenticate the client.
- OPT — Gives the client the option of submitting a certificate.
- REQ — Authenticates the client and aborts the SSL handshake if either:
 - * the client does not send a certificate, or
 - * the CA root certificate is not trusted.

This parameter is used for clients implementing SSL 3.0 only. Oracle Application Server never authenticates clients implementing SSL 2.0.

Certificate Label The label pointing to certificate files in the Security field. You can only have one certificate label per port. This allows each virtual host to use a different server certificate. If you selected NORM under Security, leave this field blank.

Configuring Server Parameters

The Server form (Figure 3-6) allows you to configure various defaults for the HTTP listener, such as default character set and time-outs.

Figure 3-6 Server form



Initial File The default file to retrieve from a directory when a URL specifies only the directory. The default filename is “`wwwIndex.html`.” For example, if you set this field to `index.html`, the request URL **`http://www.olab.com/`** will translate to **`http://www.olab.com/index.html`**.

User Directory (Unix only) The default subdirectory to search in a user’s home directory if a URL specifies only the user’s home directory. For example, if the user directory is `public_html`, and a browser requests **`http://www.olab.com/~karla`**, the URL is converted to **`http://www.olab.com/~karla/public_html/`**. There is no default.

There are several restrictions imposed on user directories:

- For security, no CGI scripts may run from a user directory.

- The HTTP listener performs no filename negotiation within user directories; the URL must specify the requested filename exactly.
- The HTTP listener does not cache files in user directories.

Note: To use this feature, the Base Directory under the Network form must be set to “/” (root). See the preceding section, [Configuring Network Parameters](#) for more information.

Default MIME Type Specifies a Multipurpose Internet Mail Extensions (MIME) type to use in interpreting requested files of an unsupported MIME type. The default is `application/octet-stream`, which means that by default, the HTTP listener treats any requested file of unidentified MIME type as a binary file. You can review the RFCs that define MIME at <http://www.oac.uci.edu/indiv/ehood/MIME/MIME.html>.

Default Character Set The character set to use in interpreting a file that uses an unrecognized character set. You must use a character set name defined by RFC 1521. The default is `iso-8859-1`: the character set used for English.

Preferred Language The language to choose when handling a request for a file available in more than one language, if the request doesn't specify a language. You must use a language identifier defined by RFC 1766. The default is “en” (English).

Imagemap Extension Filename extensions that the HTTP listener should use to identify imagemap files. There is no default, so if you want this HTTP listener to support imagemaps, you must set this field to the filename extension, such as `map`, that you use for your imagemap files.

Directory Indexing Specifies whether the HTTP listener should provide a directory listing when a URL resolves to a directory that does not contain a file named by `InitialFile`. If you enter “false,” such a URL will produce an error. Directory listings can sometimes help requestors correct spelling errors in their request URLs. The default is “true.”

If Directory Indexing is set to “true,” the listing will always be set to local time.

For security or privacy, you might not want to expose all directory contents to clients; in this case, set this field to “false.”

File Caching (Windows NT only) Determines whether the HTTP listener is going to cache a file. Possible values are:

- **ON** — Use file caching. Some editors may not be able to edit cached files. This is the default.
- **OFF** — Do not use caching.

If you want to edit files while they are being serviced by Oracle Application Server, you need to set this value to “OFF”.

Service Timeout (Windows NT only) Specifies the number of seconds the control panel services waits for a start, stop, or continue operation to finish before reporting a failure.

CGI Timeout Specifies the number of seconds the Oracle HTTP listener allows a CGI or WinCGI application to complete before terminating the script’s execution. If a CGI or WinCGI script exceeds the time limit, Oracle Application Server returns an error message without waiting for completion.

Keep Alive Timeout Specifies the number of seconds the Oracle HTTP listener keeps a connection open without any activity on that connection. If you leave this field empty, the default value is 10 seconds.

Rescan Interval Specifies the minimum amount of time (in seconds) between rescans. When a directory is accessed, the server determines when the directory was last scanned. If the directory has not been scanned within a number of seconds specified in this field, the server determines if the directory has been modified since the last scan. If it has, then the server rescans the directory.

By default, the rescan interval is set to 0, which means that a directory is checked for changes each time it is accessed. By increasing the value of this parameter, you can reduce the frequency that directories are checked for changes, which can slightly improve the performance of Oracle Application Server.

Configuring User/Group Parameters

(Unix only) The User/Group form ([Figure 3–7](#)) allows you to specify the user and group identities the HTTP listener assumes when it runs. The user and group identities that you use here must conform to existing users and groups in your Unix configuration.

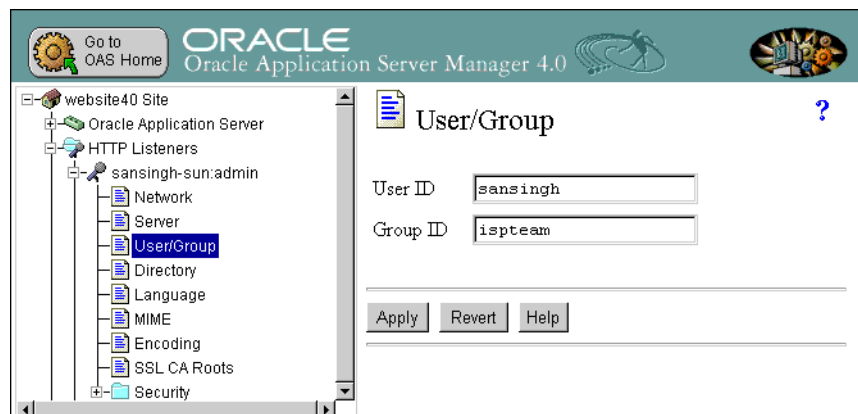
When the HTTP listener process starts up, it runs with root privileges until it completes its configuration. Then, the HTTP listener assumes the user and group identities specified in the User ID and Group ID parameters.

The User ID and Group ID parameters you use must give the HTTP listener the following access permissions to certain files and the directories in which they reside:

- Read access to the configuration file
- Read access to all files that provide content for servicing requests
- Read access to all imagemap files
- Execute access to all Oracle Application Server binaries and program files that the HTTP listener must execute
- Write access to the log and error files
- Read access to user directories

For security, however, you should beware of allowing a HTTP listener with privileged user ID, such as `root` or `oracle`, or group ID such as `dba` to launch applications, because CGI programs inherit the listener's privileges. You might want to assign your HTTP listeners an unprivileged user identity such as `nobody`.

Figure 3–7 *User/Group form*



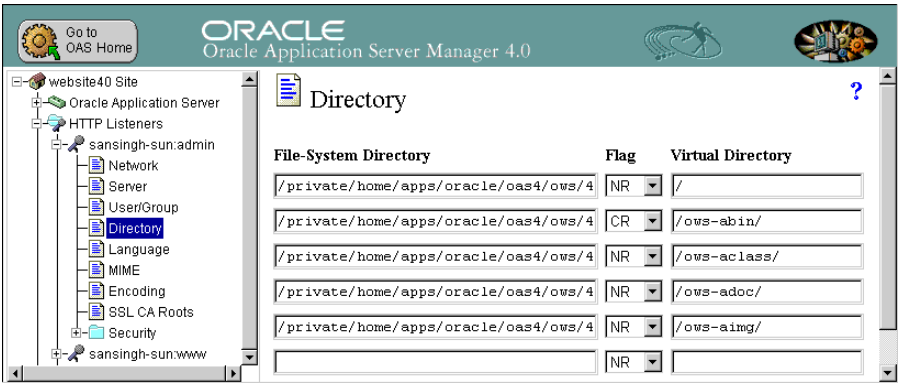
User ID Specifies the user identity that the HTTP listener assumes after it completes its configuration. You can specify either a username or numeric user ID in this field. To denote the root ID, however, you must specify `root` rather than `0`.

Group ID Specifies the group identity that the HTTP listener assumes after it completes its configuration. You can specify either a group name or a numeric group ID in this field.

Configuring Directory Mappings

The Directory form (Figure 3–8) allows you to define a virtual file system by mapping specific virtual pathnames used in URLs to the file-system pathnames of local directories.

Figure 3–8 Directory form



When the HTTP listener starts up, it checks to make sure it has access to the file-system directories specified in this section, but it does not check individual files in the directories. If a request refers to a file that the HTTP listener cannot access, the listener returns a “404: Not found” error.

Each entry in this form has the following fields:

File-System Directory Specifies the path name of a directory in the local file system. Make sure to include a trailing slash (UNIX) or a backslash (NT). For example, /privatel/oracle/app/oracle/product/7.3.2/ows/4.0/bin/.

Flag The first code in this field specifies whether CGI programs may run from the specified virtual directory. Possible values are:

- N (No CGI) — Do not allow CGI programs to run from this directory.
- C (CGI) — Allow CGI programs to run from this directory.
- S (Servlets) — Allows Java Servlets to run from this directory.
- W (WinCGI) — (Windows NT only) Allow WinCGI programs to run from this directory.

The second code specifies whether subdirectories of the specified file-system directory should be mapped recursively--that is, whether the directory tree rooted at the specified file-system directory should be accessible through the specified virtual directory. Possible values are:

- R (Recursive) — Map subdirectories recursively.
- N (Non-recursive) — Do not map subdirectories recursively.

If you do not want to make the subdirectories of the mapped file-system directory accessible to clients, set the directory mapping to non-recursive.

Virtual Directory Specifies the virtual directory through which the file-system directory will be accessed. The first entry in the Directory Mappings section must specify the virtual root directory ("/"). Make sure to include a trailing slash. For example, `/oas/oas-bin/`.

Accessing Network Mapped Directories in Windows NT

On Windows NT, in order to access a file on a network system, you must map your listener to the network path.

1. Select Services from the Control Panel and choose the listener that you want to access the network paths. Click Startup.
2. In the Service dialog box, enter a new account and password in the Log On As section. The password will not be validated until the service has been started.

Note that the HTTP listener, by default, uses the System Account. This account does not have access to files in network mapped directories.

3. Click OK.
4. Close the Services window.

To give the new user the right Log On As a Service:

1. Go to Start > Programs > Administrative Tool (Common) > User Manager. The User Manager window appears.
2. Select User Rights from the Policies pull-down menu.
3. Check the Show Advanced User Rights box.
4. From the Right pull-down menu, select Log on as a user. If the user is not listed in the Grant To box, then click Add to add the user.
5. Select the user from the Grant To box and click OK.

Configuring Language Extensions

The Language form (Figure 3–9) maps various abbreviations identifying human languages to appropriate character sets and filename extensions.

Figure 3–9 Language form

Language Type	Character Set	File Extension(s)
en	iso-8859-1	eng
en	unicode-1-1	engU uc
fr-CA	iso-8859-1	frc
fr-FR	iso-8859-1	fr

Because character set specifications apply only to text files, language extensions for non-text files are ignored. For example, the files `image.en.jpg` and `image.uc.jpg` would be considered equivalent.

Each entry in this form has the following fields:

Language Type A language identifier, such as `en`, as defined by RFC 1766.

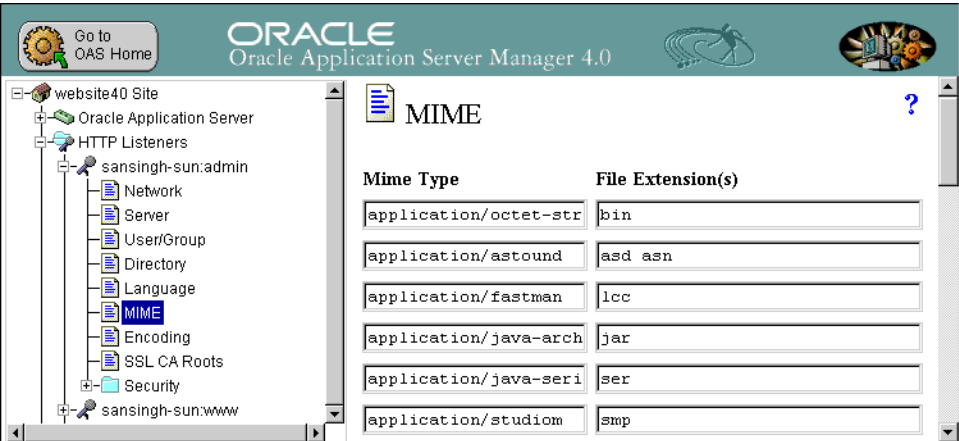
Character Set The name of the character set used for the specified language, such as `iso-8859-1`, as defined by RFC 1521.

File Extension(s) Extensions used to identify files encoded for the specified language, such as `eng`. Filename extensions are case sensitive; if you specify more than one extension, you must separate them by spaces or commas.

Configuring MIME Types

The MIME form (Figure 3-10) allows you to map Multipurpose Internet Mail Extension (MIME) types to filename extensions representing each type that the HTTP listener recognizes.

Figure 3-10 MIME form



Mime Type	File Extension(s)
application/octet-str	bin
application/astound	asd asn
application/fastman	lcc
application/java-arch	jar
application/java-seri	ser
application/studiom	smp

Each entry in this form has the following fields:

MIME Type A MIME type, such as `text/html`, as defined by RFCs 1521 and 1522.

File Extension(s) Extensions, such as `htm` or `html`, used to identify files of the specified MIME type. File extensions are case sensitive; if you specify more than one extension, you must separate them by spaces or commas. MIME extensions can be up to 8 characters long. On Unix installations, the extensions are case sensitive. On Windows NT installations, the extensions are case insensitive.

Configuring Encoding Extensions

The Encoding form (Figure 3–11) allows you to specify any processing done on a file, such as compression.

Figure 3–11 Encoding form

Encoding	File Extension(s)
compress	z
gzip	gz

Apply Revert Help

Each entry in this form has the following fields:

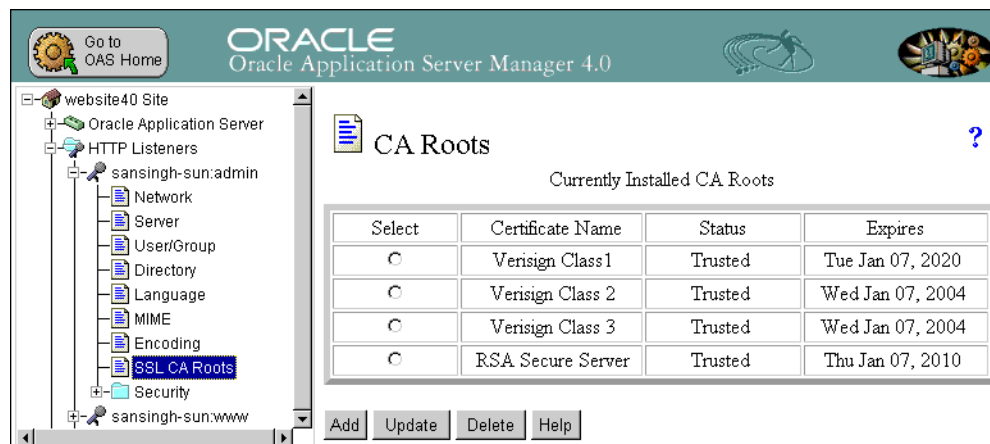
Encoding The name of a helper application, such as `compress` or `gzip`, used to encode files.

File Extension(s) Extensions, such as `z` or `gz`, used to identify files encoded by the specified helper application.

SSL CA Roots

The CA Roots form allows you to control which root certificates your website will accept. By default, Oracle Application Server trusts the four SSL CA (Certification Authority) roots shown in [Figure 3–12](#).

Figure 3–12 SSL Certificate Authentication Roots form



Status: Trusted vs. Untrusted

A “Trusted” CA root means that users who authenticate with that particular CA root will be granted access. If a CA root is “Untrusted”, those who use that CA root will be denied access.

Updating a Certificate

To change the Status of a CA root:

1. Select the certificate.
2. Click Update.
3. A window will appear with details about that particular certificate.
4. Click the button to either Trust, or Untrust this particular certificate.
5. Reload the Listener.

Adding a Certificate

If you would like to add your own certificate:

- 1. Click Add.
The CA Roots : Add form appears.
- 2. In the first box, enter a unique name for this certificate. This name must be less than 512 characters.
- 3. Paste the Base64 Certificate in the second box.

Deleting a Certificate

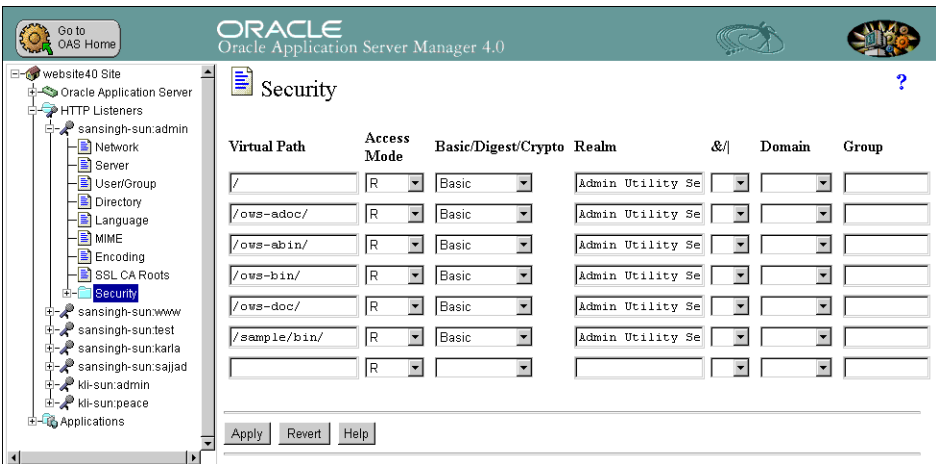
To delete an existing certificate from the list, simply select the certificate and click the Delete button.

For more information about Certificates, refer to the *Security Guide*.

Security

The Security form (Figure 3-13) allows you to assigns authentication and/or restriction schemes to protect individual files and directories.

Figure 3-13 Security form



The following fields can be configured from this form:

Virtual Path A virtual filename, directory name, or file path names incorporating a wildcard. Each HTTP listener’s virtual file system is defined on the HTTP Listener Directories form. Some examples are:

- /support/news — Specifies a single virtual filename
- /support/partners/ — Specifies an entire virtual directory
- /support/crypt* — Specifies all files in the virtual directory /support/ starting with the characters “crypt”

Note for Windows NT Users: When entering directory names into the Virtual Path field, remember to use back slashes to denote directory separations instead of forward slashes.

Access Mode A pull-down menu that specifies the access type for each set of files. You can specify a combination of the modes listed in the following table:

Access mode	How files or directories can be accessed
R	Using GET, HEAD, or POST, although the specific method may depend on the type of resource being accessed
W	Using PUT
D	Using DELETE

Basic/Digest/Crypto A pull-down menu that specifies whether to apply basic authentication, digest authentication, or Crypto SSL to the specified files or directories. If you use this field to specify an authentication scheme, you must specify a realm in the Realm field.

Realm For authentication schemes, this specifies one of the realms defined in the Basic or Digest Authentication forms.

&/| This pull-down menu allows you to assign both an authentication and a restriction scheme to the specified files or directories.

If you choose ‘&’, requestors are required to access the HTTP listener from the specified address group and provide a username and password. If you choose ‘|’, requestors may access the specified files or directories if they satisfy one of the two schemes.

Domain A pull-down menu that specifies whether to apply IP-based restriction, domain-based restriction, or neither to the specified files or directories. If you use this field to specify a restriction scheme, you must specify an address group in the Group field.

Group For restriction schemes, this specifies one of the address groups defined in the IP-based or Domain-based Restriction forms.

For more information on authentication or restriction schemes. Refer to the *Security Guide*.

Using Third-Party Listeners

You may choose to use a listener other than the included Oracle HTTP listener. Oracle Application Server supports the following third-party listeners:

- Netscape FastTrack
- Netscape Enterprise Server
- Microsoft Internet Information Server (for NT)
- Apache (for Unix)

See [Chapter 4, "Registering \(Migrating\) Third-Party HTTP Servers"](#) for more information on configuring third-party listeners.

Registering (Migrating) Third-Party HTTP Servers

Contents

- [Overview](#)
- [Registering Netscape Servers](#)
- [Registering Microsoft Internet Information Servers \(NT Only\)](#)
- [Registering Apache Servers \(UNIX Only\)](#)

Overview

A feature of Oracle Application Server Release 4.0.8 is the HTTP daemon adaptor which enables Oracle Application Server to run seamlessly using third-party HTTP servers (listeners) instead of the Oracle Web Listener shipped with Oracle Application Server. Oracle Application Server supports the migration of:

- Netscape Enterprise Server
- Apache
- Microsoft Internet Information Server (IIS)

Refer to the *Installation Guide* for the listener versions supported by Oracle Application Server.

Registering Netscape Servers

The Oracle Installer automatically detects Netscape Enterprise HTTP Servers. As part of the installation process, you can choose to register one Netscape HTTP Server automatically.

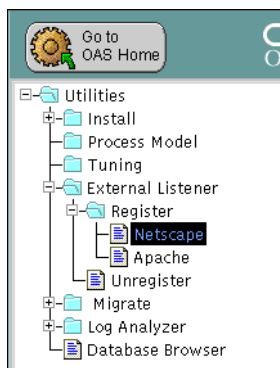
The installation automatically backs up original Netscape Enterprise HTTP server settings during the registration to **obj.conf.sav**. This file is in the same directory as **obj.conf**.

To Manually Register Netscape Server Using the OAS Utility Page

1. Go to the Oracle Application Server Welcome page at:
`http://hostname.domain:port`
(*port* is by default 8888)
and enter your username and password.
2. Click on the OAS Utilities icon.
3. Expand the tree structure under Utilities > External Listener > Register and select Netscape.

The Register Listener registration form appears.

Figure 4–1 *Netscape Listener registration form*




4. Select the node on which your listener is running.
The Netscape Listener registration form appears.

5. Enter the following information:
 - Netscape Home Directory
 - Netscape Server Type (drop-down list)
 - Admin Port Number
6. Click Search External Listeners. This will show a list of all found listeners.
7. Check the box under Register.
8. Click Register Checked Listeners.

Starting, Stopping, and Configuring the Netscape Server

Once the Netscape server has been installed, it automatically appears in the list of listeners already installed on this node. To view this list:

1. Click the  icon at the top of the page and select OAS Manager.
2. Expand the name of the web site folder in the left column and click on HTTP Listeners.

This reveals existing listeners along with the newly registered Netscape listener. From this page you can add, delete, start, stop, reload, and monitor the Netscape listener.

Changes Made During Registration

The registration process does the following:

- Configures the Netscape server configuration file **obj.conf** in the **%NS_HOME%\https-server\config** directory.
- The migration tool links these files with Oracle WRB and saves the original files into *filename.sav*.

The **init** function causes the Netscape server to load the WRB modules and adds to the **obj.conf** file as follows. Note that **<NS_HOME>** and **<ORAWEB_HOME>** will be replaced by the full path to your Netscape server home and Oracle Application Server installation directories, respectively.

```
Init fn=load-modules shlib=<ORAWEB_HOME>/lib/ndwfns362.so \
func="oracle-adp-init,oracle-adp-auth-trans,oracle-adp-service,\
oracle-adp-addlog,oracle-adp-error,oracle-adp-name-trans,\
oracle-adp-path-check,oracle-adp-object-type"
```

```
Init fn=oracle-adp-init adaptor="<ORAWEB_HOME>/lib/ndwfn362.so" \  
serverconf="<NS_HOME>/https-server/config/obj.conf" \  
servererrfile="<NS_HOME>/https-server/logs/errors" \  
mimetypesfile="<NS_HOME>/https-server/config/mime.types" listenname="mynode" \  
ORAWEB_SITE="website40"
```

Note: The file **ndwfn362.so** (**ndwfn36240.dll** for Windows NT) indicates that it is created during installation of Enterprise Server 3.6.2 for Oracle Application Server 4.0.

Additional directory mappings for the Netscape server are needed for viewing Oracle Application Server Release 4.0.8 pages. These are added to the default object section of the **obj.conf** file as follows:

```
NameTrans fn=pfx2dir from=/ows-bin dir="<ORAWEB_HOME>/bin" name="cgi"  
NameTrans fn=pfx2dir from=/ows-doc dir="<ORAWEB_HOME>/doc"  
NameTrans fn=pfx2dir from=/ows-img dir="<ORAWEB_HOME>/admin/img"  
NameTrans fn=pfx2dir from=/ows-aimg dir="<ORAWEB_HOME>/admin/img"  
NameTrans fn=pfx2dir from=/ows-adoc dir="<ORAWEB_HOME>/admin/doc"
```

The following functions are added to the **obj.conf** default object section:

```
AuthTrans fn="oracle-adp-auth-trans"  
NameTrans fn="oracle-adp-name-trans"  
PathCheck fn="oracle-adp-path-check"  
ObjectType fn="oracle-adp-object-type"  
Service fn="oracle-adp-service"  
AddLog fn="oracle-adp-addlog"
```

Registering Microsoft Internet Information Servers (NT Only)

It is important to note that Microsoft Internet Information Server (IIS) is not managed in the same way as other listeners when used with Oracle Application Server. IIS can be used with Oracle Application Server, but cannot be administered using the Oracle Application Server front end (the OAS Manager).

Starting, stopping, and configuring of IIS can only be accomplished using the Microsoft Management Console.

Enabling Microsoft Internet Information Server

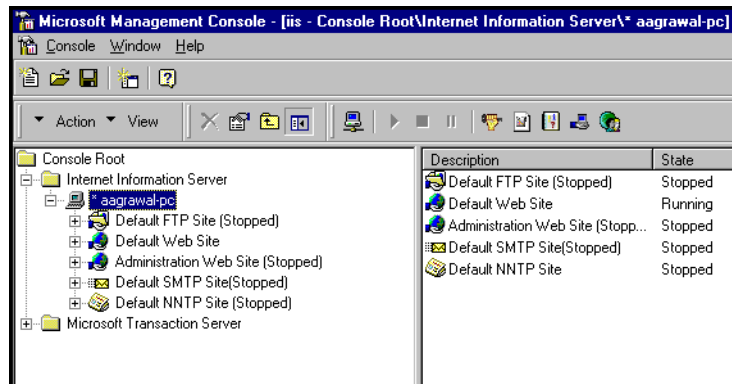
Microsoft IIS supports two modes of execution of ISAPI extensions - In-Process and Out-of-Process. **The adaptor supplied with Oracle Application Server only supports the In-Process mode.**

Using the In-Process mode, the application extension **.dll** is loaded in the IIS listener process space.

To enable the IIS listener to work with Oracle Application Server, you must use the Microsoft Management Console. In order to do this, you must add a new ISAPI filter and an associated virtual path mapping. The following steps explain how to do this:

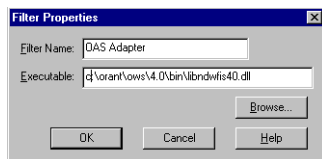
1. From the Microsoft Management Console, right-click on the name of the IIS listener you wish to configure and select Properties.

Figure 4–2 IIS Management Console



2. Select the ISAPI Filters tab and click Add. The following screen appears:

Figure 4–3 Add ISAPI filter



To add the Oracle Application Server adaptor ISAPI filter **.dll** file, enter the following information:

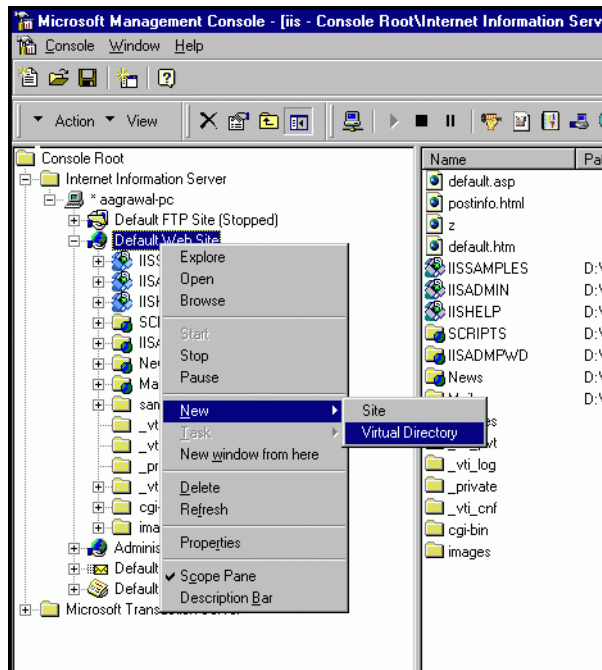
- **Filter Name:** (can be any name)
- **Executable:** %ORAWEB_HOME%\bin\libndwfs40.dll

where %ORAWEB_HOME% is the root directory in which you installed Oracle Application Server 4.0. (in the example above, c:\orant\ows\4.0)

3. Click OK, then Apply.
4. Add a new virtual path mapping for the newly created ISAPI filter.

From the Microsoft Management Console, right-click on the Web site you wish to Oracle Application Server enable (“Default Web Site” in the example below) and select New > Virtual Directory.

Figure 4–4 Add a new virtual directory



The New Virtual Directory Wizard screen appears.

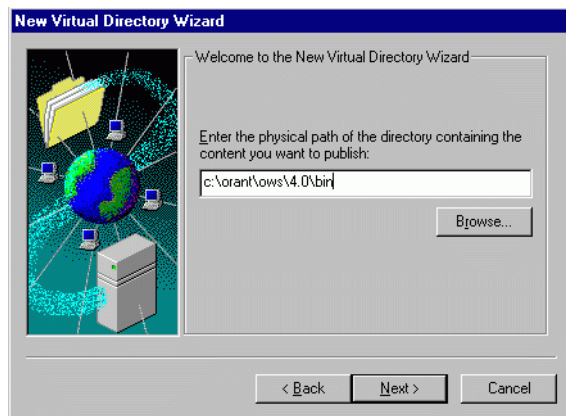
5. Enter “ows” as the alias for the new virtual directory.

Figure 4–5 Adding an alias



6. Enter the physical path for %ORAWEB_HOME%\bin.
(This is set to c:\orant\ows\4.0\bin in the example below.)

Figure 4–6 Entering the physical path



7. Set the permissions on the virtual path.
Check the box next to Allow Execute Access, then select Finish.

Figure 4-7 *Setting the access permissions*



8. Do **not** set the “Run in separate memory space” option.

Note: IIS performs lazy loading of filters. As a result, the first request to an IIS listener which requires an Oracle Application Server cartridge will take 10-15 seconds to initialize the Oracle Application Server filter.

Starting, Stopping, and Configuring IIS

To start IIS, you must first start Oracle Application Server using the OAS Node Manager, then start IIS with the Microsoft Management Console. Also, if you make a change to any of the Oracle Application Server configurations (add/delete a cartridge, add a DAD, etc.) you must reload Oracle Application Server using the Node Manager, and then stop and start IIS.

Note: The Microsoft Management Console can be used to shut-down and start IIS if the WRB is running. However, if the WRB is shut down, you cannot use the Microsoft Management Console to shut down IIS. This is a problem with IIS. To shut down IIS once the WRB has been stopped, you can either enter:

```
net stop iisadmin /y
```

at the command line or you can shutdown the service, “world wide web publishing service” from the Services window. Oracle recommends that whenever the WRB is shutdown and restarted, IIS should also be brought down and restarted.

Authentication

When using an IIS listener on NT, it is possible to perform user authentication based on the NT user domain. This is achieved by securing access to the adaptor .dll file through the validation of the client username and password against the NT user accounts.

This effectively validates access to Oracle Application Server cartridges and services at the NT listener level, and individual cartridges now cannot be protected with their own specific authentication schemes.

The benefits of this scheme are that access to an entire site encompassing static HTML pages, dynamic content such as CGI scripts and ASP application, and Oracle Application Server cartridges can all be protected with the same user domain.

Note: If the IIS authentication mechanism is used, all cartridge level authenticating must be turned off.

All requests for Oracle Application Server cartridges and services are routed through the adaptor .dll file, and therefore, you can deny/grant authentication based on the user NT domain. You can use either Oracle Application Server or IIS authentication to protect your Oracle Application Server cartridges. Disabling anonymous access causes IIS to attempt to authenticate (based on NT accounts) requests.

To use IIS authentication:

You must disable anonymous access to the “ows” virtual path. To do this, do the following:

- Do not use Oracle Application Server authentication.
- Set the variable `OAS_IIS_AUTH_ENABLED` to `TRUE`. This can be set in two places:
 - in the System Environment
 - or
 - in the Registry under `HKEY_LOCAL_MACHINE > SOFTWARE > ORACLE > OAS40`.

To use Oracle Application Server authentication:

- Allow anonymous access to the IIS.
- Either leave the variable `OAS_IIS_AUTH_ENABLED` unset, or set it to `FALSE`.
- Configure the Oracle Application Server authentication schemes.

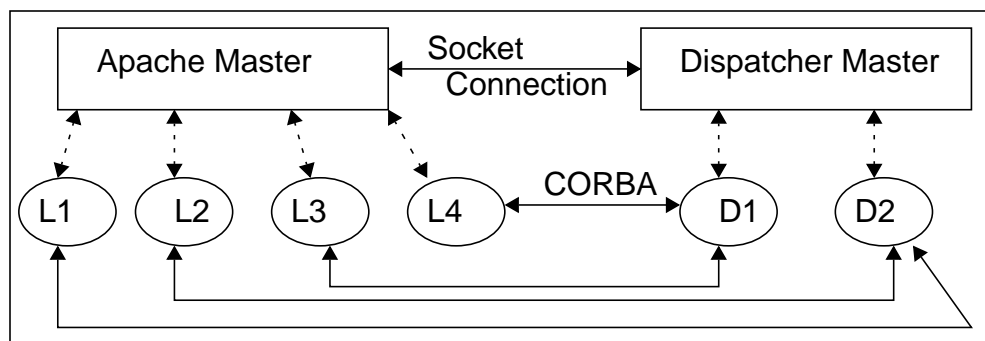
Registering Apache Servers (UNIX Only)

This section explains how to use the Apache HTTP Server with Oracle Application Server.

Apache adaptor Information

The following diagram shows the architecture of the Apache adaptor.

Figure 4–8 *Apache adaptor architecture*



The Apache HTTP Server consists of a master listener process which forks child processes (L1, L2,...) that service requests. The dispatcher is run as a separate pro-

cess and provides a thin layer in each Apache child process to allow them to talk to the dispatcher process. This is illustrated above.

Files of Interest

- **libapadp.so** - this library has most of the supporting routines required by the adaptor.
- **ndwfapd.so** - this file contains **ows_module**.

Setting up Apache

1. Compile Apache with `mod_so` support. See <http://www.apache.org/docs/dso.html> for more information.
2. Add the following line to the **httpd.conf** configuration file.

```
LoadModule ows_module ndwfapd.so
```
3. Register Apache using the OAS Utility Page. See the following section.

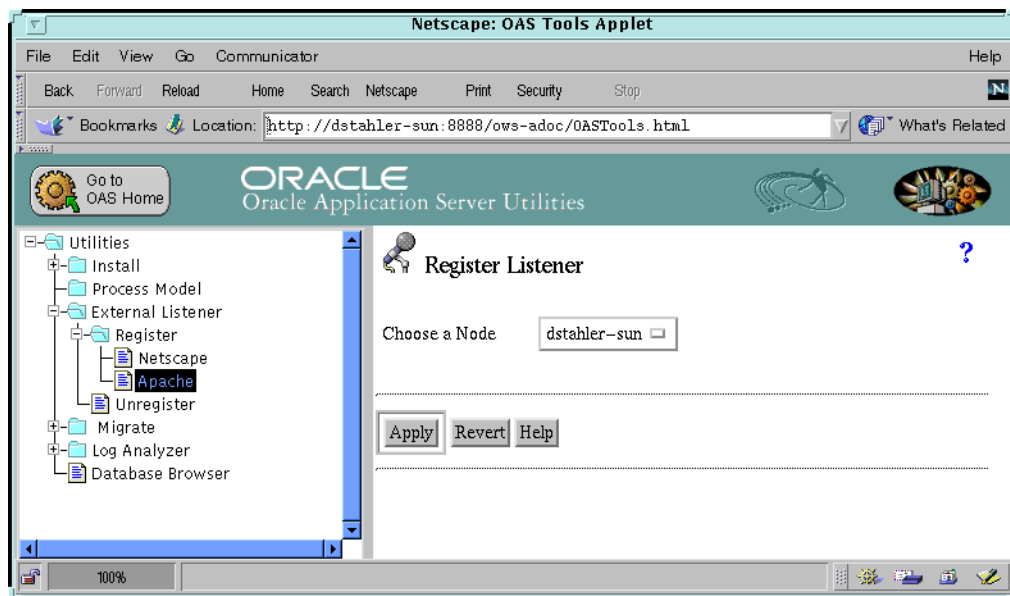
Register Apache Server Using the OAS Utility Page

1. Go to the Oracle Application Server Welcome page at:
`http://hostname.domain:port`
(*port* is by default 8888)
and enter your username and password.
2. Click on the OAS Utilities icon.

3. Expand the tree structure under Utilities > External Listener > Register and select Apache.

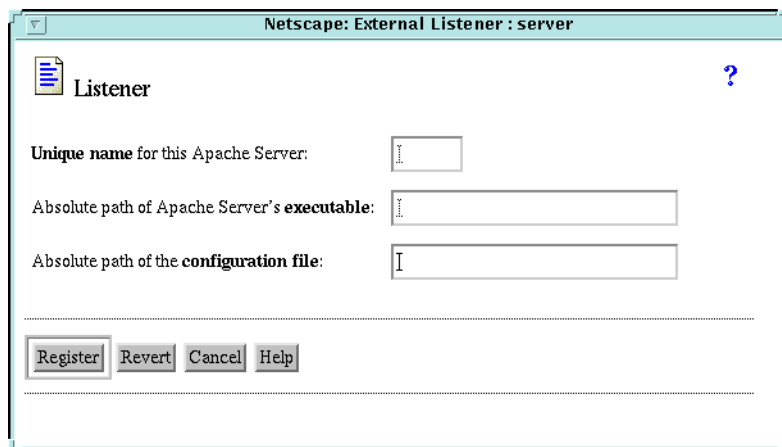
The Register Listener form appears.

Figure 4–9 *Apache Listener registration form*



4. Select the node on which the listener you want to register is running.
The External Listener form appears.

Figure 4–10 External Listener form




The screenshot shows a Netscape browser window titled "Netscape: External Listener : server". The main content area is titled "Listener" and contains three text input fields. The first field is labeled "Unique name for this Apache Server:" and has a small icon to its left. The second field is labeled "Absolute path of Apache Server's **executable**:" and has a small icon to its left. The third field is labeled "Absolute path of the **configuration file**:" and has a small icon to its left. Below the input fields is a horizontal line, followed by four buttons: "Register", "Revert", "Cancel", and "Help".

5. Enter the following information:
 - **Unique name** for this Apache Server
 - Absolute path of the Apache Server **executable** - This is the binary **httpd** file in the **bin** directory. For example, **/private/home/apache/bin/httpd**.
 - Absolute path of the **configuration file** - This is the **httpd.conf** file in the **conf** directory. For example, **/private/home/apache/conf/httpd.conf**.
6. Select Register.

Starting, Stopping, and Configuring the Apache Server

Note: If you receive a timeout error during Apache dispatcher initialization, you can increase the timeout to more than 60 seconds (which is the default). To increase the timeout period, set the environment variable `APACHE_MASTER_DISP_TIMEOUT` to a larger number (units are in seconds).

Once the Apache server has been installed, it automatically appears in the list of listeners already installed on this node. To view this list,

1. Click the  icon at the top of the page and select OAS Manager.
2. Expand the name of the website folder in the left column and click on HTTP Listeners.

This reveals existing listeners along with the newly registered Apache listener. From this page you can add, delete, start, stop, reload, and monitor the Apache listener.

Achieving Better Performance

Oracle recommends the following settings to achieve better performance with the Apache listener:

Table 4–1 Apache performance recommendations

Parameter	Recommended Setting
MaxRequestsPerChild	10000
MinSpareServers	5
MaxSpareServers	50
MaxClients	100

Running the JCO Cartridge with the Apache Server

In order to run the JCO cartridge through the Apache server, you must manually configure the Apache server configuration file to contain the virtual path `/ows-bin/` mapped to `$ORAWEB_HOME/bin`.

Object Request Broker Administration

This chapter describes basic concepts associated with the Oracle ORB.

Contents

- [Object Request Broker \(ORB\)](#)
- [Configuring ORB General Parameters](#)
- [Managing the ORB Components](#)

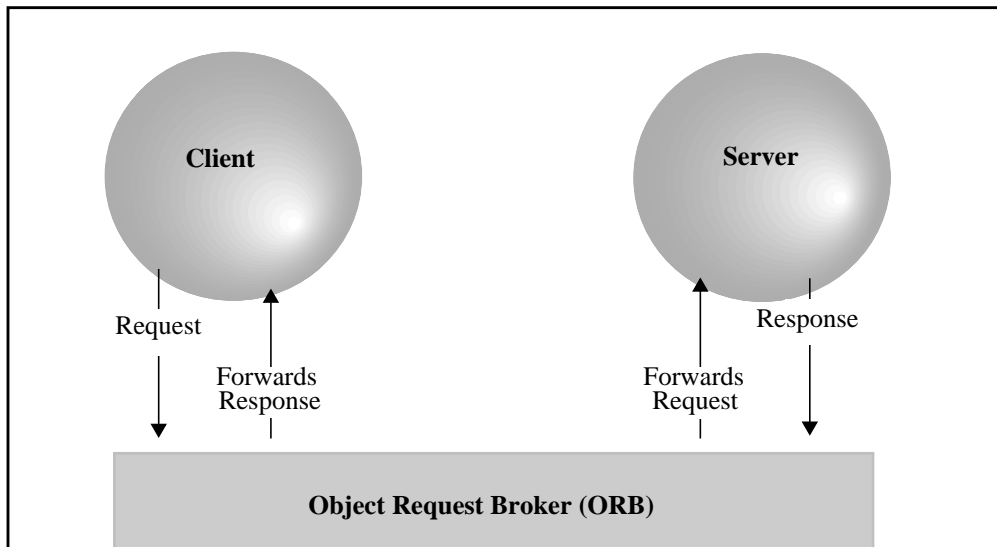
Object Request Broker (ORB)

The Oracle ORB provides a distributed-object computing environment through which clients and servers interact. Neither client or server needs to be concerned with the location of the other or the details of transporting and converting data between different computer systems. All of these details are managed by an intermediary known as the Object Request Broker (ORB). An ORB has several responsibilities, including:

- finding appropriate object implementations to handle client requests
- preparing those implementations to receive requests
- communicating the data within requests
- returning results to clients

This functionality enables clients to do business with servers regardless of factors such as the geographic locations of the servers and the programming languages in which the servers are implemented. All that a given client needs to know about a server is the server's interface.

Figure 5–1 *Client-server communication through the ORB*



Abstractly, an ORB can be implemented as a process on a client's host, as a process on some centralized computer through which client and server hosts communicate, or as a service of the operating system. The Oracle ORB uses standard CORBA IIOP protocols for communicating between host processes, one of which automatically coordinates networking traffic.

Configuring ORB General Parameters

You enter most of the ORB configuration during installation. In addition, several parameters accept defaulted values unless otherwise specified. You can change the configuration or behavior of the ORB through either the Oracle Application Server Manager.

Note: If you change any general parameters, you must stop and restart the ORB. Changing these values invalidates all outstanding object references. Any client application that has stored an object reference will receive a communication failure message when it tries to use the old object reference.

1. Using the navigational tree, expand the Oracle Application Server folder and click ORB Configuration to display the General Parameters form.

Figure 5–2 ORB General Parameters form

The screenshot shows the Oracle Application Server Manager 4.0 interface. On the left, a tree view under 'website40 Site' shows 'Oracle Application Server' expanded, with 'ORB Configuration' selected. The main area is titled 'General Parameters'. It contains two fields: 'ORB Port' with a text input containing '2649', and 'Security' with a dropdown menu set to 'OFF'. At the bottom are three buttons: 'Apply', 'Revert', and 'Help'.

2. In the ORB General Parameters form, verify, or fill in, the values:
 - **ORB Port Number**—The TCP port number on which the ORB listens for incoming connection requests. The ORB port default is 5512.
 The ORB Port Number is a TCP port number. If you change this parameter on a UNIX host, the port number must be greater than 1023, unless you are super-user or root.
 - **Security**—This parameter enables security and sets the type of access control.
 If you enable security by choosing IIOP/SSL or IIOP/IP, the next time you start up the ORB, security will be turned on. This parameter tells the ORB to load in the security dynamic libraries.
 - **OFF**—If you disable security by choosing OFF, the security dynamic libraries are not loaded. This means that the ORB footprint is smaller and any SSL client is rejected.
 - **IIOP + IP based ACLs**—Selecting IIOP/IP enables security and adds this node's IP address to the Access Control List. This node is now enabled to access ORB administration methods based upon its IP address.
 - **IIOP/SSL + certificate based ACLs**—SSL Certificate enabled. Choose to use SSL server encryption and authentication. The node is required to have a Wallet with a valid certificate to access ORB administration methods. The Wallet is created by Oracle Wallet Manager. Choosing this option adds this

node’s wallet certificate to the Access Control List. This node is now enabled to access ORB administration methods based upon its certificate.

- 3. Click Apply. Your ORB parameters have now been changed.

Managing the ORB Components

This section details how to use command-line programs to manage the Oracle ORB.

Starting and Stopping the ORB

To run ORB applications, you can start and stop ORB core services through the Oracle Application Server Manager.

Starting and Stopping the ORB


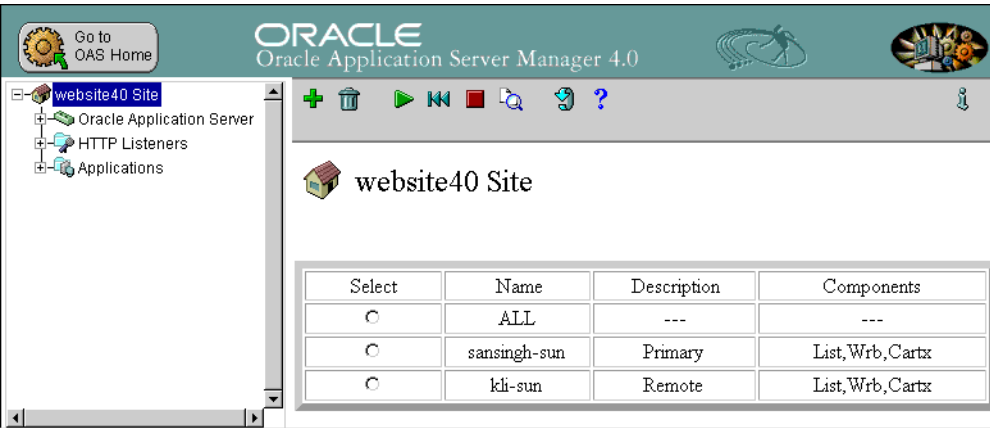


- 1. Start your browser and display the top-level administration form for Oracle Application Server. You should see “website40 Site” at the top of the left frame of the navigator (see Figure 5–3).
- 2. Click  to display the sites.


Figure 5–3 Oracle Application Server Manager form



- 3. Click “Oracle Application Server” to display the Oracle Application Server services that you can start and stop.

4. Start the ORB Servers through one of the following steps:

- Select “ALL” and click the Start icon 
- Select “orb” and click the Start icon 

Stop in the same manner, except choose the Stop icon  instead of the Start icon.

Database Access Descriptors (DADs)

This chapter contains information about Database Access Descriptors (DADs), including an overview and instructions for creating and configuring a DAD.

Contents

- [What is a DAD?](#)
- [Creating a DAD](#)
- [Configuring a DAD](#)

What is a DAD?

A Database Access Descriptor (DAD) contains the information needed by Oracle Application Server components to connect to Oracle databases. These DADs can be used by any component which supports configurable database access.

The information in a DAD includes:

- the name of the DAD
- the username to connect as
- the database user's password
- the connect-string (also known as a SQL*Net V2 connect string or Net8 service name)

The database connection information is placed in DADs so that multiple cartridges can use the same DAD. This enables you to define a DAD for each database to which you want to connect, since it is the DAD that specifies the database.

Note: For an example of how a DAD is used in conjunction with a PL/SQL cartridge, see the “PL/SQL Cartridge Overview” in *Developer’s Guide: PL/SQL Applications*.

Creating a DAD

Note: These instructions assume that the database and its listener are up and running. For more information about database concepts such as *connect string*, *username*, ***tnsnames.ora*** file, or *SID*, refer to the database documentation.

To create a DAD, use the following steps:

1. Create a connect string.

For a DAD that accesses a database on a remote host, you must specify a connect string. If your database is installed on the same host as Oracle Application Server, but is in a separate ORACLE_HOME, it is still regarded as a remote database.

The connect string must exist in the file:

- (NT) %ORACLE_HOME%\net80\admin\tnsnames.ora
- (Unix) \$ORACLE_HOME/network/admin/tnsnames.ora

where ORACLE_HOME refers to the Oracle Application Server ORACLE_HOME.

[Example 6-1](#) shows a connect string named *demo* in a ***tnsnames.ora*** file.

Example 6-1 Sample tnsnames.ora file

```
# Installation Generated Net8 Configuration
# Version Date: Nov-19-98
# Filename: Tnsnames.ora

demo =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(Host = db_machine)(Port= 1521))
    (CONNECT_DATA = (SID = orcl))
  )
```


From the example above you can see that:

- The connect string, or service name, is *demo*.
 - The protocol is *tcp*.
 - The database host (machine name) is *db_machine*.
 - The port on the database to connect to is *1521*.
 - The database name or instance is *ORCL*.
2. (UNIX users) Set the TNS_ADMIN variable to point to the **tnsnames.ora** file you just created. Verify by using the command:
 3. Test the connection to the database using the **%ORACLE_HOME/bin/tnsping** utility (where ORACLE_HOME refers to the Oracle Application Server ORACLE_HOME):

```
echo $TNS_ADMIN
```

```
tnsping demo
```

which should return a message similar to:

```
TNS Ping Utility for Solaris: Version 8.0.5.0.0 - Production on 14-APR-99 11:10:41
(c) Copyright 1997 Oracle Corporation. All rights reserved.
Attempting to contact (ADDRESS=(PROTOCOL=TCP)
(Host=db_machine)(Port=1521))
OK (20 msec)
```


4. From the Welcome page, click on OAS Manager.
 5. Expand the folder of the site you wish to configure (website40 by default).
 6. Expand the Oracle Application Server folder.
 7. Click on DB Access Descriptor.
 8. In the right frame, click on the Add  icon.
- The DAD: Add form pops up.

Figure 6–1 Database Access Descriptor (DAD) form

Netscape: Database Access Descriptor : Add

DAD : Add

DAD Name: demoDAD

Database User: demoUSER

Database User Password: ****

Confirm Password: ****

Database Location (Host): db_machine

Database Name (ORACLE_SID):

Connect String: demo

☐ Create Database User

☐ Change Database Password

☐ Store the username and password in the DAD

Apply Revert Cancel Help

Note: Some of the Utility forms enable you to create a DAD by providing a “Create New DAD” button on the bottom of the form which takes you directly to the DAD: Add form (Figure 6–1).

9. Fill in the form's fields:

DAD Name

Enter a name for the DAD you are creating. DAD names are case sensitive and can be any combination of alphanumeric characters and the “_” character.

Database User

Enter the database username that the new DAD is to assume.

Creating a New Database User (Recommended)

You can create a new user by entering the name and password in this form, and then selecting Create Database User at the bottom of the form.

Using an Existing Database User

If you elect to use an existing database user, you must perform additional tasks. After the PL/SQL Toolkit has been installed, grant the following privileges as the SYS user:

```
SVRMGR> grant ows_standard_role to <user>
SVRMGR> grant all on websys.owa_content to <user>
```

Note: Oracle Application Server does not support the use of Operating System Authentication (ops\$) accounts.

Database User Password

Enter the login password for the database user. You can either use the existing password or change it here, and select Change Database Password at the bottom of the form.

Confirm Password

Re-enter the login password to make sure you have typed it correctly.

Database Location (Host)

Specifies the machine name where the database resides. For example, john-sun.

Database Name (ORACLE_SID)

Enter the database name/instance. By default, this is usually ORCL.

Connect String

Enter the name of the connect string created in [step 1](#). In the example given, the connect string is `demo`.

For transactional DADs, it is mandatory that the connect string be specified, regardless of whether your database is local or remote.

Create Database User

Click this check-box if you are creating a new user for the database. If you have installed the PL/SQL Toolkit from this release, the new user will also be granted access to content service database objects.

Change Database User Password

Click this check-box to change the password specified in the Database User Password field.

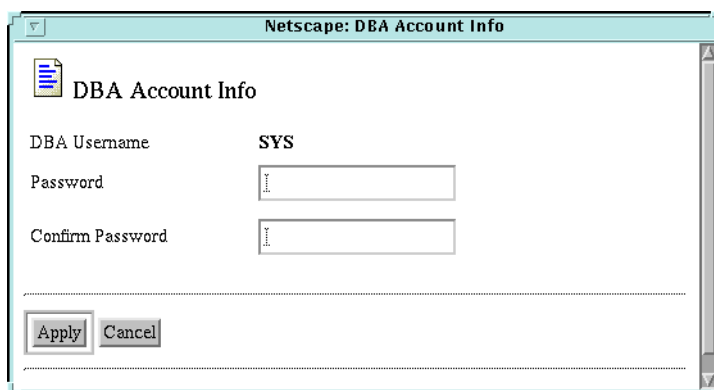
Store the username and password in the DAD

Click this check-box to store the username and password in the Database Access Descriptor. Selecting this means that you will not be prompted to enter the database username and password each time you connect to the database.

If you are going to make this a transactional DAD, and you would like the users to be able to log into different schemas using the same PL/SQL cartridge/DAD combination, make sure this option is NOT selected.

10. Click Apply.

If your database is remote, you are prompted with the DBA Account Info form.

Figure 6–2 DBA Account Info formA Netscape browser window titled "Netscape: DBA Account Info" displays a form. The form has a title "DBA Account Info" with a document icon. It contains three fields: "DBA Username" with the value "SYS", "Password" with a text input box, and "Confirm Password" with a text input box. At the bottom, there are "Apply" and "Cancel" buttons.

Netscape: DBA Account Info

DBA Account Info

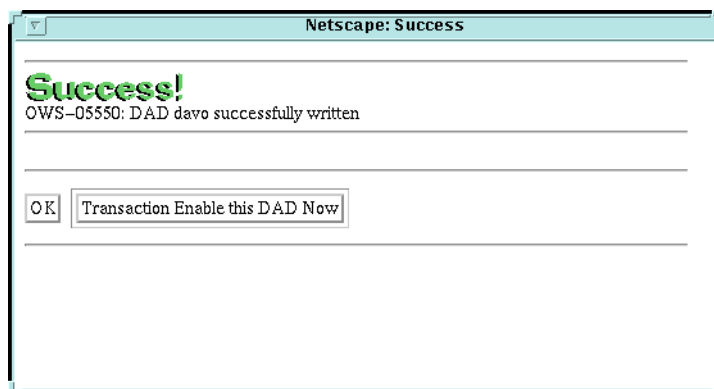
DBA Username **SYS**

Password

Confirm Password

Apply Cancel

11. Enter the **SYS** user password and click Apply.
12. If all the information is correct, a Success window pops up.

Figure 6–3 Success!A Netscape browser window titled "Netscape: Success" displays a success message. The message "Success!" is in large green text. Below it, the text "OWS-05550: DAD davo successfully written" is displayed. At the bottom, there are "OK" and "Transaction Enable this DAD Now" buttons.

Netscape: Success

Success!

OWS-05550: DAD davo successfully written

OK Transaction Enable this DAD Now

If you received an error, make sure that:

- the database is running
- the database listener is started
- the SID is correct
- the connect string matches the name entered in the **tnsnames.ora** file

- the TNS_ADMIN variable points to the **tnsnames.ora** file in your Oracle Application Server **ORACLE_HOME/network/admin** folder
- the **tnsnames.ora** file that you modified is in the Oracle Application Server ORACLE_HOME, **not** the database ORACLE_HOME

Note: If you are having problems with the TNS_ADMIN variable, reset it to the correct location and then reload Oracle Application Server.

13. Click OK.

If you wish to transaction enable this DAD, see the section, “[Adding Transactions](#)” below.

Configuring a DAD

Once you have created a DAD you can use it to connect to a database. This section contains additional DAD configuration options.

- [Adding Transactions](#)
- [Adding Advanced Features to a DAD](#)

Adding Transactions

Note: This feature is available only in the Enterprise Edition of Oracle Application Server.

You can make a DAD transactional in two ways:

- During the creation of a DAD click the “Transaction Enable this DAD Now” button (see [Figure 6-3](#)).
- After a DAD is created, select the DAD under DB Access Descriptor from the OAS Manager page and click Transactions (see [Figure 6-5](#)).

Either option takes you to the DAD: Transactions form ([Figure 6-4](#)).

Figure 6–4 DAD: Transactions

DADs that are declared transactional allow applications to participate in distributed transactions. A transaction links together multiple operations and resource changes (database updates, file changes) into a single unit, whether they occur within single or multiple applications.

Note: For more information about transactions, refer to [Chapter 11, “Enabling Transactions”](#).

Fill in the form’s fields:

- | | |
|---------------------------------|--|
| Distributed Transactions | Specifies whether to enable transactions for the DAD. Possible values are:

Enabled — Make the DAD transactional. This is the default.
Disabled — Make the DAD non-transactional. |
| Database Type | Specifies the type of database to which the DAD connects. Possible values are:

Oracle 8 — The DAD connects to an Oracle8.x database.
Oracle 7 — The DAD connects to an Oracle7.x database. |

Lock Timeout	If two requests are in contention for the Resource Manager's (RM) resources, the second request waits for the first to finish before its request is handled. This may produce a performance or deadlock situation. This value specifies the total time (in seconds) the second request is allowed to wait before it is rejected. The default is 600 seconds.
Database Link from DTC (Distributed Transaction Coordinator)	Enter a unique name for the database link. This name will be used by the DTC to connect to the DAD. Each DAD must have its own unique database link. For example, <SID>.<host-name>. This can be the global database name of the database, but need not be.
Heterogeneous OTS Agent Connect String	<p>This field is required if your RM is an Oracle7 database.</p> <p>A Heterogeneous Object Transaction Service Agent (HO Agent) is necessary to interact between an Oracle8 Distributed Transaction Coordinator (DTC) and an Oracle7 RM. If your RM is an Oracle7 database, you must configure an HO Agent to interact between them. This configuration field requires the HO Agent's service name to be configured in the DTC Oracle8 tnsnames.ora file. For more information, see Chapter 11, "Enabling Transactions".</p>

Adding Advanced Features to a DAD

You can add advanced features to a DAD, such as specifying the tablespace to use or NLS information, using the following steps.

Note: Make sure that all values for NLS-related fields are valid. The OAS Manager does not validate these fields.

1. Select the DAD to modify under DB Access Descriptor from the OAS Manager page.

Figure 6–5 Select the DAD to modify

The screenshot displays the OAS Manager interface for configuring a Database Access Descriptor (DAD). On the left, a tree view shows the hierarchy: website40 Site > Oracle Application Server > DB Access Descriptor. The 'davo' descriptor is selected. The main area on the right is titled 'Database Access Descriptor' and contains the following configuration fields:

- DAD Name:
- Database User:
- Database User Password:
- Confirm Password:
- Database Location (Hostname):
- Database Name (ORACLE_SID):
- Connect String:

Below these fields are three checkboxes:

- ☐ Create Database User
- ☐ Change Database User Password or Tablespace
- ☐ Store the username and password in the DAD

At the bottom of the configuration area are five buttons: **Apply**, **Revert**, **Advanced**, **Transactions**, and **Help**.

2. Click the Advanced button.

The DAD: Advanced form appears:

Figure 6–6 DAD: Advanced form

The screenshot shows a Netscape browser window titled "Netscape: Database Access Descriptor : Advanced". Inside the window, there is a form titled "Database Access Descriptor : Advanced" with a blue question mark icon in the top right corner. The form contains the following fields:

- Tablespace: An empty text input field.
- NLS Language: A text input field containing the value "AMERICAN_AMERICA.US7ASCII".
- NLS Date Format: An empty text input field.
- NLS Date Language: An empty text input field.
- NLS Sort: An empty text input field.
- NLS Numeric Char: An empty text input field.
- NLS Currency: An empty text input field.
- NLS ISO Currency: An empty text input field.
- NLS Calendar: An empty text input field.

At the bottom of the form, there are four buttons: "Apply", "Revert", "Cancel", and "Help".

Fill in the form's fields. Note that many of the fields are optional.

Tablespace	The tablespace to use instead of the SYSTEM tablespace. Every Oracle database contains a tablespace named SYSTEM, which Oracle creates automatically when the database is created. Make sure the parameter is valid; otherwise, you will receive a submission error or missing/invalid option.
NLS Language	Language and character set for this DAD to use, for example AMERICAN_AMERICA.US7ASCII.
NLS Date Format (Optional)	The date format to use with the TO_CHAR and TO_DATE functions, for example "MM/DD/YYYY".

NLS Date Language (Optional)	The language to use for day and month names and date abbreviations (AM, PM, AD, BC).
NLS Sort (Optional)	<p>The sorting sequence to use: BINARY or valid linguistic definition name.</p> <p>If the value is BINARY, then the collating sequence for ORDER BY queries is based on the numeric value of characters (a binary sort requires less system overhead).</p> <p>If the value is a named linguistic sort, sorting is based on the defined linguistic sort.</p>
NLS Numeric Char (Optional)	The characters to use as the group separator and decimal.
NLS Currency (Optional)	The string to use as the local currency symbol for the L number format element. For example, 'L099' represents the L format element to return the default local currency symbol for the territory FRANCE.
NLS ISO Currency (Optional)	The string to use as the international currency symbol for the C number format element. For example, 'C099' represents the C format element to return the default local currency symbol for the territory FRANCE.
NLS Calendar (Optional)	The calendar format to use since many calendar systems are in use throughout the world. For example, if NLS_CALENDAR is set to "Japanese Imperial," the date format is "YY-MM-DD."

Introduction to Applications

The Oracle Application Server is designed to run server-side applications. The application server enables you to write and to deploy your applications in several different programming languages. You can also use different protocols for clients to communicate with your applications, including HTTP or CORBA/IIOP (for details on these communications protocols, see ["Tracing Requests for Applications" on page 7-8](#)). Clients of the application server are web browsers, such as Netscape Navigator or Microsoft Internet Explorer, Java applets running within browsers, or stand-alone applications.

Contents

- [Cartridges and Applications](#)
- [Common Application Programming Models](#)
- [Tracing Requests for Applications](#)

Cartridges and Applications

Oracle Application Server provides many options for developing server-side applications; applications can be divided into the following categories:

- [Cartridge-based Applications](#)
- [IIOP Based Applications](#)

This section describes these topics.

Cartridge-based Applications

Oracle Application Server provides a powerful and efficient server-side application, called a *cartridge server*. A cartridge server is a process in which one or more cartridge instances run. A *cartridge instance* consists of code that executes application logic and configuration data that enables the cartridge to locate and to control certain parameters. In this chapter, cartridge instances are also called cartridges.

When you use the Oracle Application Server Manager to configure cartridge-based applications, you work at two levels: the application level and the cartridge level.

- Application level configuration applies configuration parameters to a collection of cartridges of the same type (refer to Table 7–1 for list of the available cartridge types). Application level configuration simplifies administration by setting common parameters for all the cartridges within an application. When the Oracle Application Server is running, it uses cartridge server processes, the runtime equivalent of applications, to create and to manage cartridge instances.
- Cartridge level configuration applies configuration parameters to individual cartridges. When the application server is running, there may be one or more instances of an individual cartridge running within a cartridge server process (the application). The cartridge level parameters apply to all the cartridge instances of a particular cartridge. When you configure a cartridge, you configure it once and all of the instances use the same configuration options.

Figure 7–1 shows a system running three cartridge servers, two for Application 1 and one for Application 2. The cartridge servers are processes that are controlled by the application server running on System A. The expanded cartridge server shown for Application 1 is running two cartridges, including: three instances of cartridge 1 and two instances of cartridge 2.

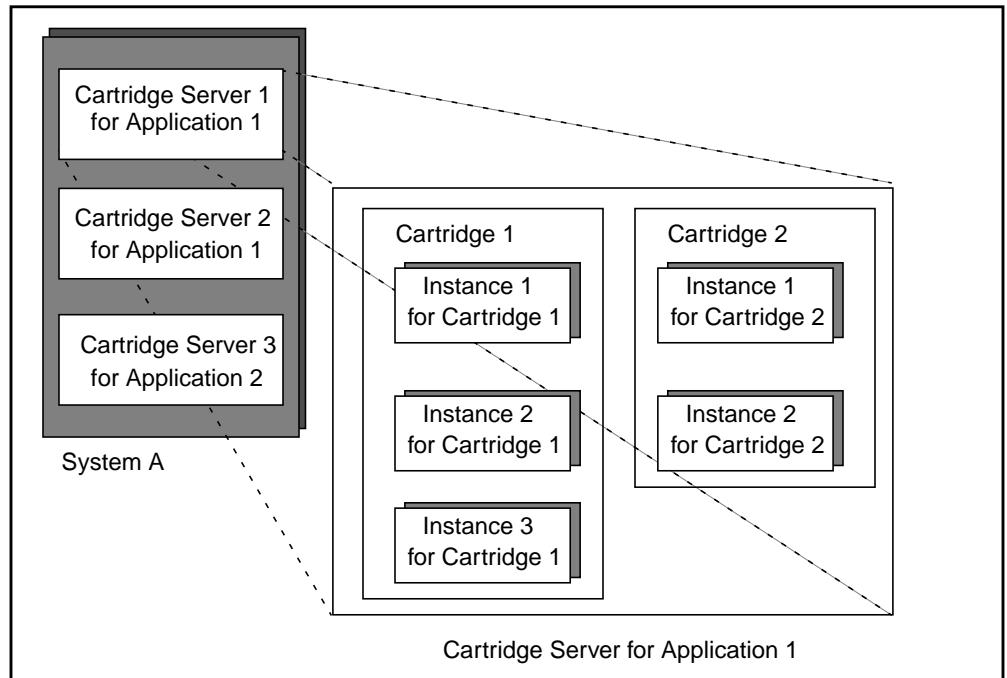
Oracle Application Server provides the cartridges listed in Table 7–1. Cartridge-based applications use the HTTP communications protocol to handle requests for clients. See the Oracle Application Server developer’s guides for specific information on how to develop cartridge-based applications.

Table 7–1 Location of cartridge based application logic

If you use:	then your application logic is in:
C cartridge	Shared libraries (UNIX) or DLL (NT)
JServlet, JWeb cartridges	Java class files
LiveHTML cartridge	embedded scripts in HTML files

Table 7–1 (Cont.) Location of cartridge based application logic

If you use:	then your application logic is in:
ODBC cartridge	Procedures in ODBC databases such as Sybase or Informix
Perl cartridge	Perl scripts
PL/SQL cartridge	Stored procedures in databases or SQL script files.

Figure 7–1 Cartridge instances and cartridge server processes

IIOP Based Applications

Oracle Application Server allows you to develop applications using distributed objects based on the CORBA standard with communication based on the IIOP protocol. The application server supports the following application models:

- [ECO/Java Applications](#)
- [Enterprise JavaBeans \(EJB\) Applications](#)
- [C++ CORBA Cartridge Applications](#)

ECO/Java and EJB let you to create CORBA objects that can be accessed from different types of clients such as Java applets, Java applications, and CORBA applications. For details on the design of ECO/Java and EJB applications, refer to the *Developer's Guide: ECO/Java and EJB Applications*.

Since ECO/Java and EJB applications are handled differently at the cartridge runtime level, as compared to cartridge-based applications, the Oracle Application Server Manager displays *application instances* rather than cartridges for these configuration options. The application server uses ECO/Java and EJB server processes, the runtime equivalent of applications, to create and to manage application instances.

Using the Oracle Application Server Manager to control and configure ECO/Java and EJB applications consists of administration tasks that are similar to those for cartridge-based applications. The cartridge level interface and the configuration options for ECO/Java and EJB components have different names and different parameters; these differences are required to support the IIOP transport and to improve performance (refer to the section [“Configuring ECO/Java and EJB Components”](#) in [Chapter 9, “Cartridge and Component Administration”](#) for information on configuring ECO/Java and EJB applications).

When you use the Oracle Application Server Manager to configure ECO/Java and EJB applications, you work at three levels: the application level, the application instance level, and the object level.

- Application level configuration applies configuration parameters to a collection of application instances of the same type (the types currently supported are ECO/Java and EJB). Application level configuration simplifies administration by setting common parameters for all application instances within an application.
- Application instance level configuration applies configuration parameters to objects within the application instance. When the application server is running, there may be one or more instances running within a server process (the application). The application instance level parameters apply to all the instances.
- Object level configuration applies configuration parameters to individual objects within an application instance.

ECO/Java Applications

In the ECO/Java model you deploy components written in Java as CORBA components. These components, called ECO/Java objects are Java classes that you package together to form ECO/Java applications running in an Oracle Application

Server environment. An ECO/Java application consists of one or more ECO/Java objects.

Instances of ECO/Java objects can be instantiated and accessed from different types of CORBA clients such as JServlet applications, Java applets, Java applications, Web cartridges, and CORBA applications.

CORBA clients, such as Java applets running on browsers or Java applications, use ORBs to obtain object references for ECO/Java object instances. Once a client has an object reference to an ECO/Java object instance, it can invoke methods on the instance.

You would generally deploy this kind of application with a client applet, or a web front end. In this model, the client uses a web browser to download and run the applet. The applet uses the ORB built into the browser to communicate with the Oracle Application Server application using IIOP (see [Figure 7-3](#)).

Enterprise JavaBeans (EJB) Applications

Enterprise JavaBeans (EJB) is a standard introduced by JavaSoft that enables developers to create custom component applications. These applications consist of enterprise beans developed by your company or by third parties. The beans provide the business logic in EJB applications. Oracle Application Server provides support for EJB applications.

In Oracle Application Server, EJB applications run in a CORBA environment. This means that the beans themselves are CORBA objects and can communicate with other CORBA objects. The containers for the beans are also CORBA objects, and they interact with other components of the application server.

C++ CORBA Cartridge Applications

The C++ CORBA cartridge defines a component architecture for building distributed object-oriented business applications in the C++ programming language. It addresses the development, deployment, and runtime aspects of application development. The C++ CORBA cartridge makes it easy for application developers to write applications. The cartridge runtime shields the developers from low-level details of transactions, multi-threading, resource pooling, etc.

The C++ CORBA cartridges are created and managed at runtime by the C++ CORBA cartridge container, which is provided by Oracle Application Server. The characteristics of the cartridge, such as timeout values, state, etc. are customized at the time of deploying the application. A consistent view of the cartridge is given to the client regardless of how the C++ cartridge is implemented and what functions it provides to the client.

A C++ application consists of one or more C++ cartridges. C++ cartridges typically provide the business logic in C++ applications. The cartridges define methods that clients can invoke to perform some operation.

Common Application Programming Models

The architecture of Oracle Application Server lends itself to three basic approaches to application design:

- [The Request-Response Model](#)
- [The Session Model](#)
- [The Transaction Model](#)

When you configure applications, it is helpful to know what type of cartridge or ECO/Java/EJB component you are installing or modifying. The developer who implements a cartridge or component should know which of these approaches applies to the application.

The Request-Response Model

Applications using the request-response model respond to each client request individually. After the application responds with the requested content, all data associated with the client and the request is lost. This is consistent with the “statelessness” of ordinary HTTP requests.

Typically, an application of this model identifies the request, responds with the requested content, and returns without saving any state. A request for an HTML page is an example of the request-response model.

The Session Model

Applications using the Session model in the application server maintain a persistent association between clients and particular cartridge instances of the application. Subsequent requests from the same client are handled by the same cartridge instance. The session persists until the application has been idle for a configurable time-out period or until the application breaks the association.

Applications of this model save client-related state data in an application context structure. Because the session mechanism guarantees an association between a client and a cartridge instance, the application behaves as if it is interacting with a single client. For example, if you maintain a client-specific counter in your application context structure, the value of the counter may be different for each cartridge instance, and is correctly maintained for each client. In contrast, if your application

uses the request-response model, such a counter is shared by all cartridge instances in a cartridge server.

As another example, if you are developing a human resources application that enables users to enter and update personal information, you can use sessions in the application to keep track of user data. As the user navigates from page to page, the cartridge instance can display the data that the user has entered so far. The cartridge instance contains data from one user only.

Note: Using either the request-response or the session model, you can save client data in the application context structure of the cartridge instance. If you are using the request-response model, however, it does not make sense to do so because the next request from the same client may go to another cartridge instance.

The Transaction Model

Applications that allow users to purchase products typically use the transaction model. When the user pays for the products, the transaction service is used between the payment and shipment systems. The transaction is committed only when both systems approve the request; if one or both systems reject the request, the transaction is aborted.

Applications in this model typically perform database transactions that span multiple HTTP requests. These applications use the Transaction service APIs, which are available only in the Enterprise Edition of the application server. See [Chapter 11, “Enabling Transactions”](#) for more information on the Transaction service APIs.

When a transaction application receives a request, it checks if the request is to begin a transaction. If so, it starts a transaction and subsequent requests are executed within the context of that transaction. The application determines when to commit or roll back the transaction.

Note: Applications in the transaction model can be request-response or session oriented. Requests from clients do not have to be directed to the same cartridge instance.

Tracing Requests for Applications

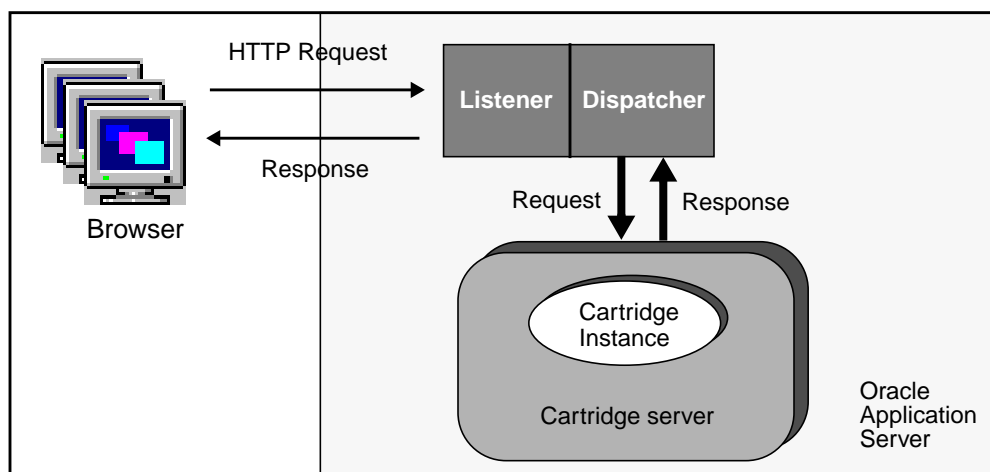
Oracle Application Server can handle both [HTTP Requests](#) from web browser clients, and [CORBA/IIOP Requests](#) from CORBA clients by way of a client ORB.

HTTP Requests

[Figure 7-2](#) shows the sequence of events when a web browser client issues an HTTP request to a server-side application.

The dispatcher maintains a cache of available cartridge instances. When a request arrives for a particular cartridge, the dispatcher routes the request to one of its cached cartridge instances of the appropriate type (See ["Configuring Tuning Parameters" on page 9-17](#) for information on limitations on the number of available cartridge instances).

Figure 7-2 *Life cycle of an HTTP request*



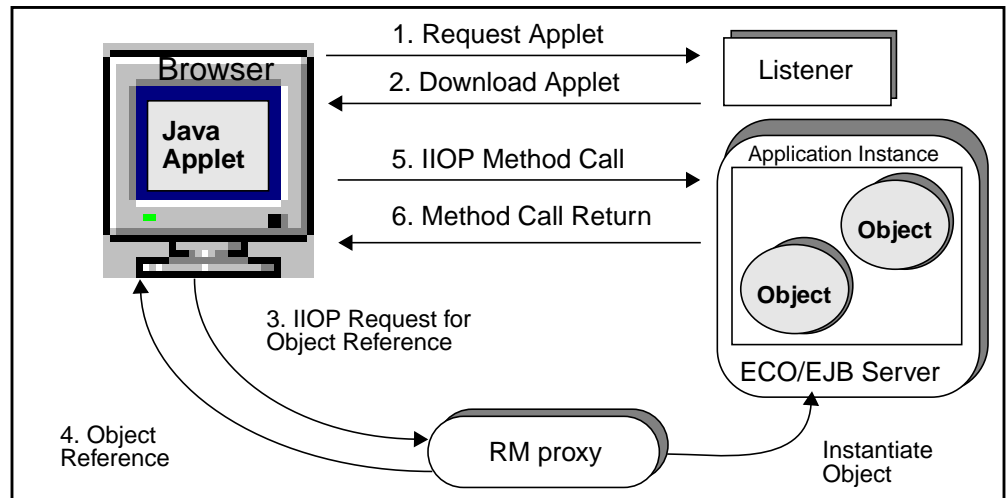
CORBA/IIOP Requests

[Figure 7-3](#) shows the sequence of events when a Java applet issues a CORBA/IIOP request to a server-side ECO/Java or EJB object.

When a client requests an object reference, the RM proxy instantiates the requested object, if necessary, within a CORBA server and returns the reference to the client. The client then uses the object reference to call methods on the object directly.

The life-cycle of a request for an EJB application is similar to that of an ECO/Java application. The RM proxy creates an instance of the specified EJB object in an EJB server. For more information on CORBA/IOP requests, see the *Oracle Application Server Developer's Guide: ECO/Java and EJB Applications*.

Figure 7-3 Life cycle of a CORBA/IOP request



When a client requests an object reference, the RM proxy instantiates the requested object, if necessary, within a CORBA server and returns the reference to the client. The client then uses the object reference to call methods on the object directly.

The life-cycle of a request for an EJB application is similar to that of an ECO/Java application. The RM proxy creates an instance of the specified EJB object in an EJB server. For more information on CORBA/IOP requests, see the *Oracle Application Server Developer's Guide: ECO/Java and EJB Applications*.

Application Administration

The Oracle Application Server Manager allows you to work with forms to manage and configure applications. This chapter describes the application-level configuration forms and the procedures you use to add, configure, and manage applications.

Contents

- [Concepts](#)
- [Application Configuration Roadmap](#)
- [Adding and Configuring Applications](#)
- [Managing Applications](#)

Concepts

When you work with the Oracle Application Server Manager, you can configure and manage two types of applications:

- [Cartridge-based Applications](#)
- [ECO/Java and EJB Applications](#)

There are different forms and different options depending on the type of application that you need to add, configure, or manage. How you work with the Oracle Application Server Manager depends on the types of applications that you run on your Web site.

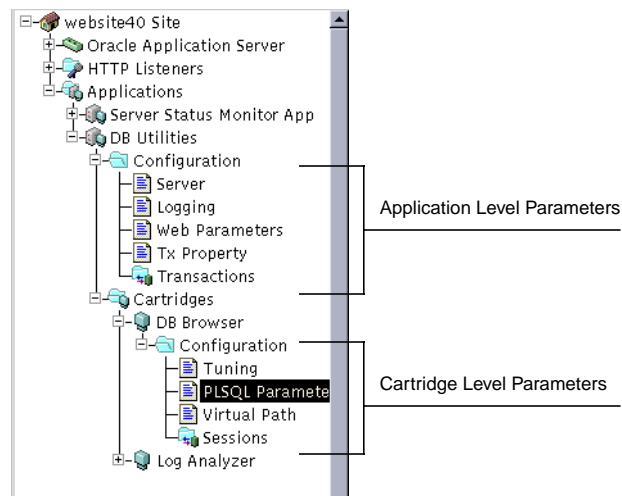
Cartridge-based Applications

When you use the Oracle Application Server Manager to configure cartridge-based applications, you work with two sets of configuration parameters:

- [Application Level Parameters](#)
- [Cartridge Level Parameters](#)

[Figure 8–1](#) shows the configuration options for a PL/SQL application and a cartridge within the application.

Figure 8–1 Configuration parameters for a PL/SQL application



Application Level Parameters

Using the Oracle Application Server Manager, application level parameters are located in the configuration folder under each application. [Table 8–1](#) describes some of the parameters you can set using the application configuration forms. There are several configuration forms that are common to most Oracle Application Server applications: Server, Logging, Web Parameters, and Transaction related forms. Some applications have additional configuration forms.

Application level configuration simplifies administration by setting common parameters for all the cartridges within an application. When the Oracle Application Server is running, it uses cartridge server processes, the runtime equivalent of applications, to create and manage cartridge instances.

Cartridge Level Parameters

Cartridge level parameters are located in the configuration folder under each cartridge. [Figure 8–1](#) shows the “DB Utilities” application that has two cartridges: “DB Browser” and “Log Analyzer”. The configuration folder shows the forms available for these cartridges. Cartridge level forms let you apply configuration parameters to individual cartridges. When the application server is running, there may be one or more instances of an individual cartridge running within a cartridge server process (the application).

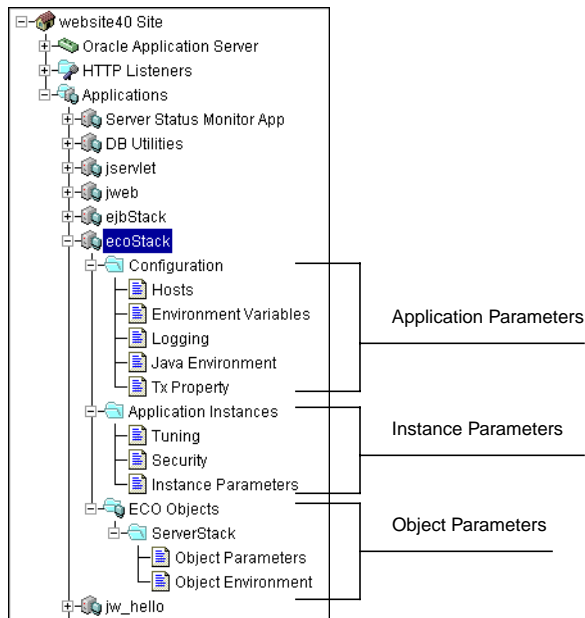
The configuration options for cartridges vary depending on the cartridge type. If you are adding a PL/SQL application, for example, the application’s cartridges have different configuration forms from cartridges in a Perl application. Refer to [Chapter 9, “Cartridge and Component Administration”](#) for details on adding and configuring cartridges.

ECO/Java and EJB Applications

When you use the Oracle Application Server Manager to configure ECO/Java or EJB applications, you work with three types of configuration parameters:

- [Application Level Parameters](#)
- [Application Instance Parameters](#)
- [Object Parameters](#)

[Figure 8–2](#) shows the configuration options for an ECO/Java application and its application instances.

Figure 8–2 Configuration parameters for a ECO/Java application

Application Level Parameters

Using the Oracle Application Server Manager, application-level parameters are located in the Configuration folder under the application name. [Table 8–1](#) describes some of the parameters you can set using these application configuration forms. There are several configuration forms that are common to most Oracle Application Server applications: Hosts, Logging, and Transactions. Some applications have additional configuration forms.

Application Instance Parameters

Application instance parameters are located in the Application Instances folder for each ECO/Java or EJB application. There is a configuration form for each ECO/Java or EJB application instance. Refer to [Chapter 9, "Cartridge and Component Administration"](#) for details on adding and configuring ECO/Java and EJB application instances.

Object Parameters

ECO/Java and EJB component object parameters affect the ECO/Java or EJB components running within the ECO/Java or EJB server process. Object configuration

forms are located in the ECO/Java Objects or EJB Objects folder (object folders are named for each object). Refer to [Chapter 9, "Cartridge and Component Administration"](#) for details on ECO/Java and EJB Object configuration.

Table 8–1 Application-level parameters

To set this...	Use this parameter	... in this form	Default
How long (in seconds) a cartridge instance can be idle before it is terminated	Cartridge timeout	Server (see Configuring Cartridge Server Options)	86400
Logging parameters	Logging	Logging (see Configuring Logging Parameters)	On
MIME types that the application handles	Application MIME Types	Web Parameters (see Configuring Web Parameters)	All except jpeg and gif
Whether or not clients need certificates in order to access the application	Client certificate	Web Parameters (see Configuring Web Parameters)	Disabled
Whether or not each cartridge instance is bound to a client	Client sessions	Web Parameters (see Configuring Web Parameters)	Disabled
The number of seconds before a session times out	Max session idle time	Web Parameters (see Configuring Web Parameters)	600 for Jservlet 15 otherwise
The page that is returned to the client if an error occurs	Error page	Web Parameters (see Configuring Web Parameters)	no default
Enable transaction service for the application	Transactions	Tx Property (see Chapter 11, "Enabling Transactions").	not transactional
The transactional database access descriptors used by the application	Transactional DADs	Tx Property (see Chapter 6, "Database Access Descriptors (DADs)").	no configured DADs are available

Note: There are additional parameters that are set at the cartridge level for cartridge-based applications, and at the application instance level for ECO/Java or EJB applications. Refer to [Chapter 9, "Cartridge and Component Administration"](#) for details on these parameters.

Application Configuration Roadmap

Once your web site nodes and HTTP listeners are configured, and you have developed applications to install on your Oracle Application Server, you can add and configure the applications. The following gives you a checklist for adding and configuring applications.

- Configure the access descriptor to any databases that your application uses. See [Chapter 6, "Database Access Descriptors \(DADs\)"](#) for directions on configuring a database access descriptor (DAD).
- Add the application See [“Adding Applications” 8 on page 8-8](#) for directions on adding an application.
- Configure the application using the application level forms. The application level configuration applies parameters across all cartridges or application instances. See [“Configuring Applications” 14 on page 8-14](#) for directions on configuring the application.
- Add and configure cartridges for cartridge based applications.

Configure the application's cartridges. For cartridge based applications you need to add and configure one or more cartridges. See the section [“Configuring Cartridges” 16 on page 9-16](#) in [Chapter 9, "Cartridge and Component Administration"](#) for directions on configuring cartridges.

Cartridge configuration sets general parameters, such as cartridge name, display name, tuning parameters and the virtual path for all of a cartridge's cartridge instances. In addition, each cartridge type has distinct configuration requirements, as listed in the following table:

Application Type	Cartridge Configuration Requirements
C++ CORBA	Environment Variables
PL/SQL	DAD name for destination database
JServlet	Default
JWeb	Default and Java environment variables
LiveHTML	LiveHTML parameter form
Perl	Perl parameter form
CWeb	Cartridge entry point, Cartridge parameter form, CWeb parameter form.

- Configure application instances and objects for ECO/Java or EJB applications. You can use the following forms to configure CORBA applications:

- Application Instance form
- Object Parameters form

See [“Configuring ECO/Java and EJB Components”](#) **30** on page 9-30 in [Chapter 9, “Cartridge and Component Administration”](#) for directions on configuring your EJB or ECO/Java application instances.

- Configure the ORB. All applications use the ORB internally for communication between the client browser and the Oracle Application Server. You can change the defaults for the ORB and for the security used by the ORB. See [Chapter 5, “Object Request Broker Administration”](#) for directions on configuring ORB parameters.
- Configure Transactions. If your application is transactional, you need to configure your database to be transactional. See [Chapter 11, “Enabling Transactions”](#) for directions.
- Configure Security. If your application requires encryption, authentication or other security features, you need to configure security. See the Oracle Application Server *Security Guide* for details.
- Configure Logging. If you desire logging for your application, see [Chapter 10, “Logging and Database Utilities”](#) for directions on how to enable logging.

Adding and Configuring Applications

An application consists of one or more cartridges which contain the application logic, or for ECO/Java or EJB applications, the application consists of one or more programs which define Java components (OAS runs Java Virtual Machines that contain the Java components). Adding applications to the Oracle Application Server requires that you perform several procedures using the Oracle Application Server Manager.

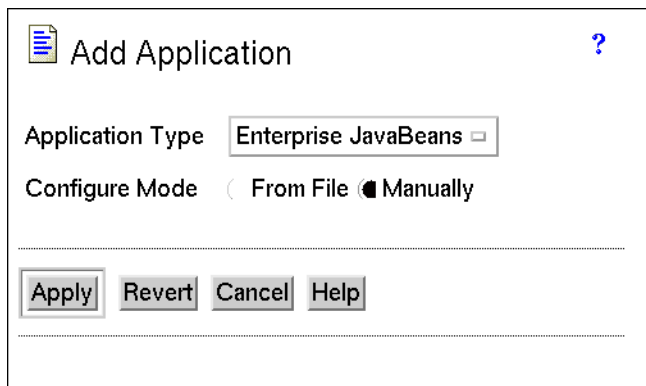
- Add the application according to the instructions in this section.
- Configure the application according to the instructions in this section.
- Add and configure one or more cartridges for cartridge based applications according to the instructions in [Chapter 9, “Cartridge and Component Administration”](#).

- When you add applications, you have to reload the application server for the new configuration to take effect. Refer to [“Reloading Applications”](#)³¹ on [page 8-31](#) for information on reloading the application server.

Adding Applications


The Oracle Application Server Manager Add Application form lets you add a new application (see [Figure 8-3](#)).

Figure 8-3 Add Application form



The screenshot shows a web-based form titled "Add Application". The form is enclosed in a rectangular frame. At the top left of the frame is a small icon of a document with lines, and at the top right is a blue question mark. Below the title, there are two main sections. The first section is labeled "Application Type" and contains a pull-down menu currently showing "Enterprise JavaBeans". The second section is labeled "Configure Mode" and contains two radio buttons: "From File" and "Manually". The "Manually" radio button is selected. At the bottom of the form, there is a horizontal line, and below it are four buttons: "Apply", "Revert", "Cancel", and "Help".

Display the Add Application form by performing the following steps:

1. Connect to the Oracle Application Server Manager using your web browser (see ["Accessing the Welcome Page"](#) in [Chapter 1, "Overview"](#) for details).
2. Click on the "+" next to a site name to display the components on the site (for example, website40). In the navigation tree, you should see "Oracle Application Server", "HTTP Listeners", and "Applications".
3. Select "Applications" to display the applications in the right frame.
4. On the applications page in the right frame, click . This brings up the Add Application form shown in [Figure 8-3](#).
5. In the Add Application form make selections for the following:
 - Application Type: this pull-down menu allows you to select the new application type.

- **Configure Mode:** select Manually or From File. The Manually option lets you to enter configuration options using dialog boxes. The option, From File, assumes that configuration information for the new application is available in a file (see on page 8-12).

Note: EJB and ECO/Java applications must be added manually.

- Click “Apply”.

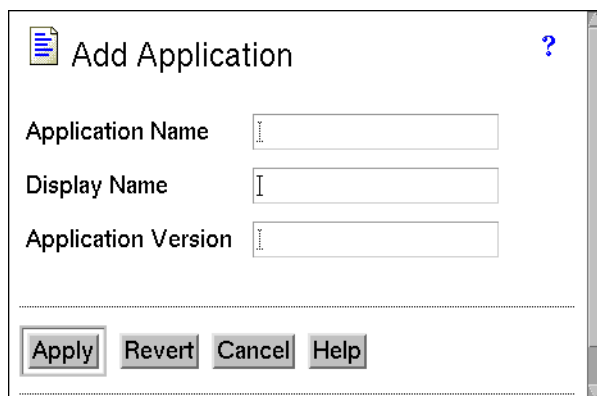
This displays the manual Add Application form. Refer to the application specific sections listed in [Table 8-2](#) for details on adding an application of the specified type.

Table 8-2 Add Application Form options

Application Type	Refer to Section
C++ CORBA	Adding a C++ CORBA Application
CWeb	Adding a Cartridge-Based Application
Enterprise JavaBeans	Adding an EJB or ECO/Java Application
ECO/Java	Adding an EJB or ECO/Java Application
JServlet	Adding a Cartridge-Based Application
JWeb	Adding a Cartridge-Based Application
LiveHTML	Adding a Cartridge-Based Application
Perl	Adding a Cartridge-Based Application
PL/SQL	Adding a Cartridge-Based Application

Adding a Cartridge-Based Application

[Figure 8-4](#) shows the cartridge Add Application form for the CWeb, JServlet, JWeb, LiveHTML, Perl, and PL/SQL applications. Refer to the following sections for the dialogs for information on adding C++ CORBA, ECO/Java and EJB applications.

Figure 8–4 Cartridge Add Application form

The screenshot shows a standard Java Swing-style dialog box titled "Add Application". The title bar includes a small icon of a document with a question mark. The main area of the dialog contains three text input fields, each with a label to its left: "Application Name", "Display Name", and "Application Version". Below these fields, there is a horizontal line separating them from a row of four buttons: "Apply", "Revert", "Cancel", and "Help". The "Apply" button is highlighted with a darker border.

The Add Application form contains three fields: Application Name, Display Name, and Application Version. After you fill in the fields and select “Apply”, the Oracle Application Server Manager displays a confirmation dialog. This dialog presents two buttons: “OK”, and “Add Cartridge to this Application”.

- “OK” confirms your addition of the new application and adds the application.
- “Add Cartridge to this Application” allows you add a cartridge to the new application. For details on adding cartridges, and using the add cartridges form, refer to [Chapter 9, "Cartridge and Component Administration"](#).

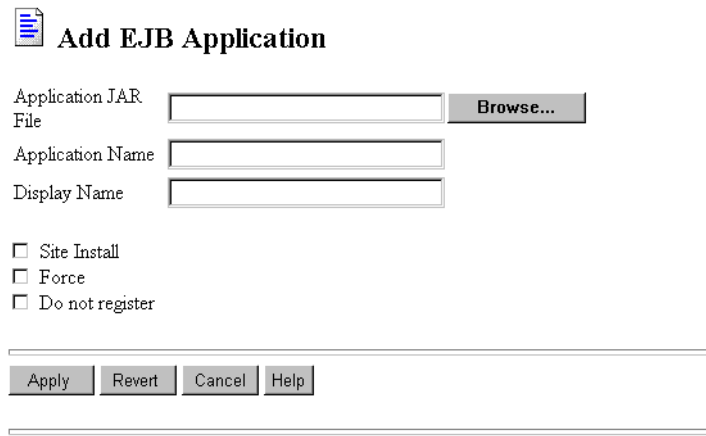
Application Name Enter the name for the application. Do not use commas, periods, or spaces as part of the application name. Sixty characters (60) is the maximum length permitted for the application name.

Display Name Enter the display name for the application. This is the name that appears on the Oracle Application Server Manager navigational tree.

Application Version Enter the application version for the application. For example, 1.0.

Adding an EJB or ECO/Java Application

[Figure 8–5](#) shows the Add Application form for EJB applications. (The Add Application for ECO/Java applications is identical.) The form contains three fields: Application Jar File, Application Name, and Display Name.

Figure 8–5 Add EJB Application form


Add EJB Application

Application JAR File **Browse...**

Application Name

Display Name

☐ Site Install
☐ Force
☐ Do not register

Apply **Revert** **Cancel** **Help**

Application JAR File Enter the pathname for the ECO/Java or EJB JAR file. The Browse button allows you to browse through the file system to select the JAR file.

When adding the application, the JAR file will be transferred to the server automatically.

Note: The JAR file cannot be loaded from the server machine.

Application Name Enter the name of the application. Do not use commas, periods, or spaces as part of the application name. Sixty characters (60) is the maximum length permitted for the application name.

Display Name Enter the display name of the application. This is the name that appears on the navigational tree.

The Add Application forms for ECO/Java and EJB applications contain three checkboxes that allow you to select the following options:

Site Install	This checkbox allows you to install the application's jar file on the primary node, and on all available remote nodes.
Force	This checkbox allows you to reuse a name to replace an existing jar file and all of its configuration information.

Do Not Register This option replaces the jar file without registering the application. This is helpful when you are replacing an existing application's jar file with a new jar file and you do not want to replace the existing configuration information.

After you fill in the fields and select “Apply”, the Oracle Application Server Manager displays a confirmation dialog.

Adding a C++ CORBA Application

The Add Application form shows the fields you need to enter to add a C++ CORBA application (see [Figure 8–6](#))

Figure 8–6 Add C++ CORBA Application form

Application Config File

Enter the name of the file containing the C++ CORBA configuration information.

Application Library

Enter the full path to the library .so file (UNIX) or .dll file (NT) that contains the application logic for the cartridge

Force This checkbox allows you to reuse a name to replace an existing jar file and all of its configuration information.

Adding an Application from a Configuration File

[Figure 8–7](#) shows the Add Application form for adding an application using configuration information from an existing .app file. For information on creating an

`<application>.app` file, see the Oracle Application Server *Cartridge Management Framework* manual.

Figure 8–7 File Add Application form

Configure File For the Configure File field, enter the file system path to the `<application>.app` file containing the configuration information for your application. For example, `$ORACLE_HOME/ows/cartx/plsql/install/myapplication.app`. For more information about the `<application>.app` file, see the Oracle Application Server *Cartridge Management Framework* manual.

Adding an Application to a Remote Node

If you are running Oracle Application Server on multiple nodes and you want to add an application to a remote node, you need to install the application on the remote node. This section covers the procedures for adding an application to a remote node.

Adding a Cartridge-based Application to a Remote Node In order to add a cartridge-based application to a remote node, you must first add the application to the primary node and then configure the application to run on the remote node.

The following steps outline the procedure for adding a cartridge based application to a remote node.

1. Add the application to the primary node as described in this section.
2. Add a cartridge or several cartridges to the application on the primary node (refer to [Chapter 9, "Cartridge and Component Administration"](#)).
3. Copy all of the cartridge files to the remote node or nodes. Note that you do not need to copy the files to all remote nodes, but only to the remote nodes where you want to run the application.

For example, if the application resides in the directory \$ORACLE_HOME/../../cartx/livehtml/samples on the primary node, then copy the files to \$ORACLE_HOME/../../cartx/livehtml/samples on the remote node(s).

Note: The directory where you place the application files on the remote node(s) must be the same as on the primary node.

4. Configure the application to use the remote node by selecting the remote node on the Hosts form (refer to [“Configuring Hosts for Load Balancing” 19 on page 8-19](#) for information on the Hosts form).

Adding an ECO/Java or EJB Application to a Remote Node In order to add an application to a remote node, you must first add the application to the primary node and then deploy the application to the remote node.

Adding an C++ CORBA Application to a Remote Node In order to add an application to a remote node, you must first add the application to the primary node and then configure the application to run on the remote node.

Configuring Applications

This section describes the configuration forms that control application level parameters. Application level configuration parameters apply to all cartridges within a cartridge based application. For ECO/Java or EJB applications, the application level configuration parameters apply to all application instances within the application.

When you install a new application or when you modify an application, you have to either reload or stop certain components, such as listeners so that the changes take effect. There are two types of configuration changes:

- **Changes to Application Logic** – for example, when you update a PL/SQL application, or supply a new ECO/Java application containing new Java class files.
- **Changes to Application, Cartridge, or Component Configuration**– for example, you may want to add or remove cartridges from an application, change the logging levels, or change tuning parameters. Configuration changes affect the Oracle Application Server wrb.app configuration file. Whether you need to stop or reload the application for the changes to take effect depends on the configuration parameter that you modify.

Changes to Application Logic


[Table 8–3](#) shows the components you have to stop when you make a change to cartridge or component application logic.

Table 8–3 *Modifying applications and affected components*

Type	If you change this:	You have to stop this:
C	The library file (.so or .dll)	Cartridge server
ECO/Java or EJB	Java classes for the ECO/Java or EJB components.	ECO/Java or EJB server (you also have to re-install the application)
JServlet or JWeb	Java classes	Cartridge servers that invoke the class
LiveHTML	Files called by the LiveHTML cartridge	Nothing
LiveHTML	Perl scripts called by a LiveHTML cartridge	Cartridge servers that invoke the script
Perl	Perl scripts called by the Perl cartridge	Nothing
PL/SQL	Procedure in the database	Cartridge instances that invoked the procedure

Note: When you stop cartridges or application instances you do not restart them using the Oracle Application Server Manager. The application server starts cartridge servers when it receives a request or to comply with the current minimum number of cartridge servers set in the tuning parameters.

Changes to Application, Cartridge, or Component Configuration

[Table 8–4](#) lists the configuration parameters that can be reloaded. If you change a parameter found in this list, you can reload the application and the change takes effect. Reload an application by selecting  from the Oracle Application Server Manager. If the parameter is not in this list, you have to stop the application.

Many Oracle Application Server Manager application configuration changes require that you that stop and restart listeners. For example, when you make changes that affect an application's virtual path, or other changes that require you

to stop or reload the cartridge servers for an application, you have to stop and restart the listeners for the change to take effect (refer to Table 8–4 for a description of the actions that require that listeners be stopped and restarted).

If you make modifications that cannot be reloaded, you have to stop and restart the application server (see [“Stopping Processes and Applications”](#)³³ on page 8-33 for information on stopping and restarting the application server).

Note: Whenever you stop and restart Oracle Application Server components, you also have to stop and restart the listeners.

Table 8–4 *Reloadable parameters*

Action	Effects
Adding an application	The resource manager (RM) creates the minimum number of cartridge servers for new applications. You have to stop and restart the listener.
Removing an application	The RM shuts down all existing cartridge servers for the removed application. You have to stop and restart the listener.
Adding a cartridge to an existing application	The RM creates the minimum number of instances of the added cartridge. You have to stop and restart the listener.
Removing a cartridge from an application	The RM removes all instances of the removed cartridge from all cartridge servers. You have to stop and restart the listener.
Modifying the minimum or maximum number of cartridge servers and cartridges	<p>The RM enforces the maximum value for cartridge servers immediately if the new maximum value is greater than the previous maximum value.</p> <p>If the new maximum value for cartridge servers is less than the previous value and the current server count is greater than the new maximum value, the RM enforces the new maximum value after all cartridge instances in a cartridge server are terminated.</p>
Adding or removing hosts for an application	<p>The RM stops cartridge servers from hosts that are removed.</p> <p>The RM uses the new host list when it checks the number of cartridge servers for the application.</p>

Table 8–4 (Cont.) Reloadable parameters

Action	Effects
Modifying load-balancing information	The RM uses the new configuration when it checks the number of cartridge servers for the application.
Modifying log parameters	The new configuration goes into effect for the next logged message.

Configuration Forms

Application configuration tasks involve working with forms using the Oracle Application Server Manager. The following forms are available for application configuration. Depending on the application type, some of the configuration forms may not be used. Forms that are not used are not shown in the Oracle Application Server Manager.

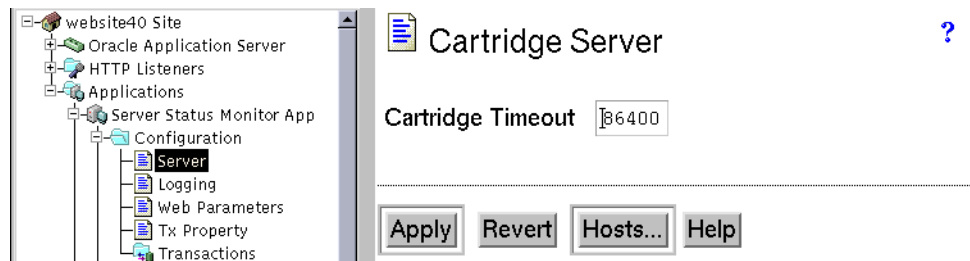
- [Configuring Cartridge Server Options](#)
- [Configuring Logging Parameters](#)
- [Configuring Web Parameters](#)
- [Configuring Transactions](#)
- [Configuring Environment Variables](#)
- [Configuring the Java Environment](#)

Note: Depending on the type of the application and the setup of your site, you may not need to make any modifications to the default configuration when you add an application, or you may need to work with several of the application configuration forms.

Configuring Cartridge Server Options

Figure 8–8 shows the Cartridge Server form which allows you to configure the Cartridge timeout and other cartridge server related parameters. A cartridge server runs one application, which consists of one or more cartridges.

Figure 8–8 Cartridge Server form (Min/Max load balancing scheme)



Cartridge timeout

When a cartridge instance has been idle for the specified number of seconds, the cartridge server closes the connection and terminates the instance. This behavior is independent of whether the instance is session-enabled (for details on session enabled cartridges, see "[Client sessions](#)" on page 8-24).

Changing the Application Priority Level

When you use priority based load balancing, the cartridge server form provides an additional field that lets you select the application priority. Using priority based load balancing, you can set and change your application priority level from the default value, Medium, to a value that suits your needs. **Figure 8–9** shows the Cartridge Server form shown for priority based load balancing. See the Oracle Application Server *Performance and Tuning Guide* for more information on choosing a load balancing scheme.

Figure 8–9 Cartridge Server form (Priority based load balancing)

The screenshot shows the 'Cartridge Server' configuration window. On the left is a tree view of the application hierarchy: website40 Site > Oracle Application Server > HTTP Listeners > Applications > Server Status Monitor App > Configuration > Server. The 'Server' node is selected. The main panel on the right is titled 'Cartridge Server' and contains the following fields:

- Cartridge Timeout (in seconds):** A text input field containing the value '186400'.
- Priority:** A dropdown menu currently set to 'Medium'.

At the bottom of the form are four buttons: 'Apply', 'Revert', 'Hosts...', and 'Help'.

Priority Defines the priority level for the application. The form only shows this option when you are using priority based load balancing (see [Figure 8–9](#)). Resources are allocated to the application based on the priority level that is set for the application. All cartridges contained in this application use the specified priority unless the cartridge overrides the value set at the application level.

Configuring Hosts for Load Balancing

The Hosts form allows you to select which hosts to run the application's cartridge servers or the ECO/Java or EJB application. [Figure 8–10](#) shows the Hosts form. You can access this form by clicking on the "Hosts" button on a cartridge based application's Cartridge Server form or by selecting the Hosts from the application Configuration folder in an ECO/Java or EJB application (see [Figure 8–8](#) or [Figure 8–9](#)).

You only need to work with the Hosts form when you are running the Oracle Application Server on multiple nodes. If you have installed the application on remote nodes, then you can use the Hosts form to specify where the application runs. For details on installing an application on a remote node, refer to ["Adding an Application to a Remote Node" 13 on page 8-13](#).

Figure 8–10 Cartridge Server Hosts form (Min/Max load balancing)

Select	Host	Initial Start-up Instances
<input checked="" type="checkbox"/>	system1	<input type="text" value="10"/>
<input checked="" type="checkbox"/>	pdsun-dev1	<input type="text" value="10"/>

Apply Revert Cancel Help

The Hosts form contains the following fields:

Field	Comments
Select	A check-box option to select a particular host.
Host	Displays the host name. For example, <code>is-system31</code> .
Initial Start-up Instances	<p>This field is only shown for Priority based load balancing. Specifies the minimum number of cartridge servers that run on the node.</p> <p>When the application server starts up, it starts up the minimum number of cartridge servers. As it receives requests beyond what the minimum number of cartridge servers can handle, it starts up more cartridge servers, up to the maximum number.</p>
Min # of servers	Specifies the minimum number of cartridge servers that can run on the node. See Max # of servers for details. This field is only shown for Min/Max based load balancing.
Max # of servers	<p>Specifies the maximum number of cartridge servers that can run on the node. This field is only shown for Min/Max based load balancing</p> <p>When the application server starts up, it starts the minimum number of cartridge servers. As it receives requests beyond what the minimum number of cartridge servers can handle, it starts up more cartridge servers, up to the maximum number.</p>

Each application can service one or more clients, depending on the number of cartridge instances and threads. See [Chapter 9, "Cartridge and Component Administration"](#) for more details on the cartridge instances and threads parameters.

When you are running Oracle Application Server on multiple nodes, the minimum number of cartridge servers is not spawned immediately when you start the application server. The application server enforces the specified minimum number of cartridge servers only after allowing some time for the cartridge server factories on all the remote nodes to start up. The cartridge server factories start up cartridge servers. The time allowed is five minutes.

For more information about load balancing and performance issues, see the Oracle Application Server *Performance and Tuning Guide*.

Configuring Logging Parameters

[Figure 8–11](#) shows the form which allows you to configure Logging parameters for your application. See [Chapter 10, "Logging and Database Utilities"](#) for details on configuring and working with the Logger.

Figure 8–11 *Logging form*

The screenshot displays the Oracle Application Server configuration interface. On the left, a tree view shows the hierarchy: website40 Site > Oracle Application Server > HTTP Listeners > Applications > Server Status Monitor App > Configuration > Server > **Logging**. The right pane, titled 'Logging', contains the following configuration options:

- Logging:** A dropdown menu set to 'ON'.
- Logging Directory:** A text input field containing '...'.
- Logging File:** A text input field containing 'I'.
- Severity Level:** A dropdown menu set to '0'.

At the bottom of the right pane are three buttons: 'Apply', 'Revert', and 'Help'.

Logging

The Logging parameter specifies whether you can log messages from your application. Possible values are:

- ON – Log messages. This is the default.
- OFF – Do not log messages.

Logging Directory

The parameter Logging Directory specifies the pathname for the log file directory.

By default this is set to the pathname, %ORAWEB_ADMIN%/website40/log. To change the default value, specify a complete directory pathname. For example, /private/OAS/logs/.

Logging File

The Logging File parameter specifies the log file name. The default logging file is wrb.log (in the directory %ORAWEB_ADMIN%/website40/log).

Severity Level

The Severity Level parameter specifies the severity level of messages that are logged. Only messages with a severity level lower than or equal to the specified severity level are logged. Specifying a high value causes the logger to log more messages. For example, if you set the severity level at 3, the logger logs messages of severity levels 0, 1, 2, or 3. Refer to [Chapter 10, "Logging and Database Utilities"](#) for a detailed description of the severity levels.

Configuring Web Parameters

[Figure 8–12](#) shows the Web Parameters form. The Web Parameters form is only supplied for applications that use the HTTP communication protocol (ECO/Java and EJB applications do not use the HTTP communication protocol). See “Tracing Requests for Applications” on page 8 in [Chapter 7, "Introduction to Applications"](#) for information on communication protocols.

Figure 8–12 Web Configuration Form

The screenshot displays the 'Web Configuration' form. On the left, a tree view shows the hierarchy: 'website40 Site' > 'Oracle Application Server' > 'HTTP Listeners' > 'Applications' > 'Server Status Monitor App' > 'DB Utilities' > 'Configuration' > 'Web Parameters' (highlighted). The right pane contains the following settings:

- Application MIME Types:** A text field containing 'jpeg,gif'.
- Client Certificate:** A button labeled 'Disabled' with a small square icon.
- Client Sessions:** A button labeled 'Disabled' with a small square icon.
- Max session idle time:** A text field containing '15'.
- Error Page:** An empty text field.

At the bottom of the form are three buttons: 'Apply', 'Revert', and 'Help'.

Application MIME types

This field specifies a list of file extensions that the application does or does not process.

To list the supported file extensions, prefix the list with a + character (for example, "+ html, txt").

To list the file extensions that the application does not process, prefix the list with a "-" character (for example, "- jpeg, gif").

If you do not enter a value, then the cartridge accepts all file extensions and all URLs are sent to the cartridge.

Client certificate

When this parameter is set to enabled, Oracle Application Server requires clients to have an appropriate SSL certificate to access the application. The possible values are:

- Disabled – Do not retrieve the client-side SSL certificate for the cartridge. Disabled is the default.
- Enabled – Retrieve the client-side SSL certificate for the cartridge.

Client sessions

The parameter lets the dispatcher know that the cartridge is session enabled; requests originated from the same client receive the same cartridge instance (See *Overview of Oracle Application Server Overview and Glossary* for information on the dispatcher). This is true as long as the idle time is less than the specified time in the Max session idle time field. The possible values are:

- Disabled – The cartridge is not session-enabled. This is the default.
- Enabled – The cartridge is session-enabled. If you select this option, you also need to enter a value for the Max session idle time field.

Disabling the client sessions parameter does not mean that an application cannot maintain state information in a session. Instead, it means that the Oracle Application Server does not manage the state information for your application. State information can still be managed using the session programmatic interface. Refer to Chapter 4, “Advanced JServlet Programming,” in the *Oracle Application Server Developer's Guide: JServlet Applications* for more information about using sessions with JServlet applications.

Max session idle time

This parameter specifies the time, in seconds, before a session between a client and a cartridge instance that is session-enabled times out.

Once the idle time exceeds the time specified in this field, the cartridge instance is no longer associated with the request and can be bound to any other incoming request. After the instance expires, the client asking for the expired cartridge receives a page back indicating that the session expired. The subsequent request starts a whole new session.

Use the Max session idle time field with caution. If you set this field to a very large number, during that time, the cartridge is unavailable to any other request. The optimal value is dependent on the functionality of the cartridge. Set it to a value that is large enough to allow multiple requests within the session to complete, but small enough so that it does not stop the cartridge from being used by other requests unnecessarily.

The default for this field is 15 except for JServlet applications.

For JServlet applications, the default value is 600. Oracle Application Server ignores this value when the `setMaxInactiveInterval` method is used in the application. See the *Developer's Guide: JServlet Applications* for more information.

Error page

Specifies the page to display if an error occurs in executing a request. This page appears when a client session has expired and the client tries to resume the session.

This field is optional. If no page is specified, **\$ORAWEB_HOME/admin/doc/wrberr.html** is used.

Configuring Transactions

The transactions forms allows you to enable applications for transactions. Only the following applications support transactions:

- PL/SQL
- JServlet
- EJB
- ECO/Java
- C++ CORBA
- LiveHTML
- JCORBA
- CWeb
- JWeb

For detailed information about building transactional applications, refer to [Chapter 11, "Enabling Transactions"](#).

Configuring Environment Variables

[Figure 8–13](#) shows the Environment Variables form that allows you to define application environment variable values. You should not remove any of the values from the pre-defined environment variables, but you can define your own environment variables or append values to the pre-defined variables.

If you need to enter additional environment variables, and you run out of empty lines on the form, click “Apply”. This saves the information you enter and gives you additional blank lines.

[Table 8–6](#) describes the standard environment variables found in the Environment Variable form. [Table 8–6](#) describes additional environment variables found with Java-based applications.

Figure 8–13 Environment Variables form

Environment Variables	
CLASSPATH	%ORAWEB_HOME%/classes/services
LD_LIBRARY_PATH	/private/oracle01/app/oracle/p
PATH	%ORAWEB_HOME%/jdk/bin:%PATH%
JAVA_HOME	%ORAWEB_HOME%/jdk
THREADS_FLAG	native

Table 8–5 Standard Environment variables

Variable	Description
LD_LIBRARY_PATH	(UNIX only) A colon-separated list of directories that contain libraries.

Table 8–6 Environment variables for Java-based applications

Variable	Description
CLASSPATH	<p>A list of directories or jar files that contain class files for your objects. For example: %ORAWEB_HOME%/classes/ejbapi.jar.</p> <p>When you install your application, CLASSPATH is set to access all classes contained in the jar file being deployed as well as other supporting classes.</p> <p>On Unix platforms, directories and jar files in CLASSPATH are colon-separated (i.e., first.jar:second.jar). On Microsoft Windows NT, directories and jar files in CLASSPATH are semicolon-separated (i.e., first.jar;second.jar).</p>
PATH	<p>A list of directories (colon-separated on Unix, semicolon-separated on NT) that contain executables. This should be set to contain %ORAWEB_HOME%\jdk\bin.</p>

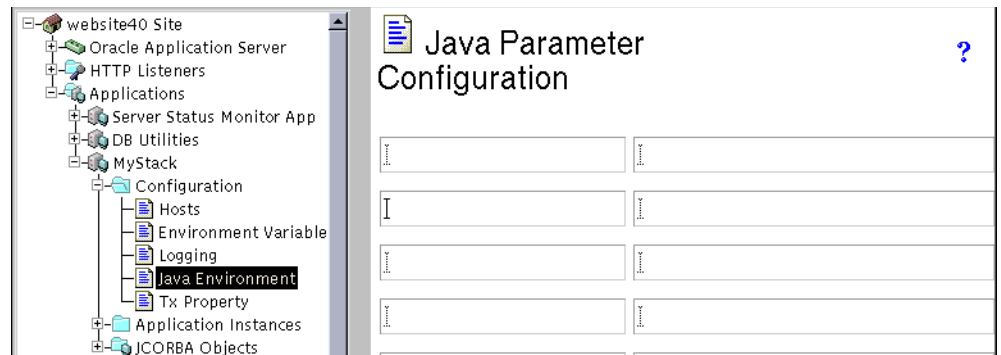
Table 8–6 (Cont.) Environment variables for Java-based applications

Variable	Description
JAVA_HOME	The top-level directory where Java is installed. This should be set to %ORAWEB_HOME%\jdk.
THREADS_FLAG	(UNIX only) Whether the Java Virtual Machine should use native threads or not. This is set to “native”. This value is required.

Configuring the Java Environment

Figure 8–14 shows the Java Environment form that allows you to define Java properties. You should not remove any of the values from the pre-defined properties, but you can define your own Java properties or append values to the pre-defined properties.

If you need to enter additional Java properties, and you run out of empty lines on the form, click “Apply”. This saves the information you enter and gives you additional blank lines.

Figure 8–14 Java Environment form

Managing Applications

Using the Oracle Application Server Manager you can manage applications by performing the following tasks:

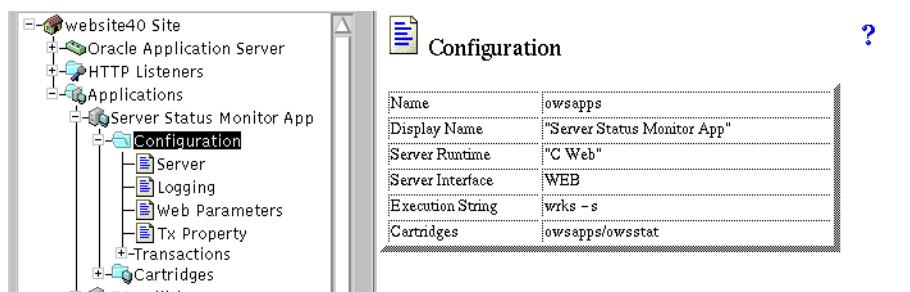
- [Displaying Configuration Information](#)
- [Monitoring Running Applications](#)
- [Renaming Applications](#)

- [Reloading Applications](#)
- [Deleting Applications](#)
- [Stopping Processes and Applications](#)
- [Stopping and Restarting Listeners](#)

Displaying Configuration Information

Figure 8–15 shows the Configuration form listing application information.

Figure 8–15 Configuration form



To display the configuration information, perform the following steps:

1. In the navigational tree, expand Applications by selecting the “+” next to Applications in the tree. This displays a list of all the applications.
2. Expand the application of interest, for example, the Server Status Monitor Application.
3. Select the Configuration folder for the selected application.

Table 8–7 describes the fields in the application information form. Some of the fields are only shown for applications of a particular type. For example, the Client Jar file field is only shown for EJB applications.

Table 8–7 Application Information form fields


Field	Description
Name	Displays the short name for the application.
Display Name	Displays the complete name for the application. This is the name that is used in the navigational tree.

Table 8–7 (Cont.) Application Information form fields

Field	Description
Server Runtime	Displays the server runtime.
Server Interface	Displays the server interface. Valid values are Web and CORBA.
Application Version	Displays the version of the application.
Execution String	Displays the execution string for the application. For example, wrks -s.
Client Jar File	Displays the complete path to the client jar file.
Server Jar File	Displays the complete path to the server jar file.
Application File	Displays the complete path to the application's .app file.
Cartridges	Displays the complete cartridge name running in the application. For example, MyApplication/MyCartridge.

Monitoring Running Applications

The Applications Monitoring form displays statistics and status information for selected applications. To monitor applications perform the following steps:

1. In the navigational tree, select the application that you want to monitor. This displays a list of cartridges in the application.
2. Select the button next to “ALL”, or select the cartridges that you want to monitor.
3. Click  to monitor the application or the selected cartridges.

This presents the screen shown in [Figure 8–16](#) (the information shown is application specific).

Note: All the monitoring statistics are a rough indication of the state of the server. They are not precise.

Figure 8–16 Applications Monitoring form

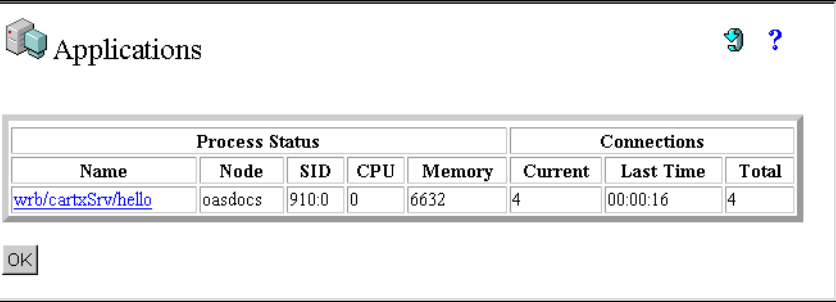


Table 8–8 and Table 8–9 describe the Monitor display fields.

Table 8–8 Monitor process status information

Column	Description
Name	The name of the process.
Node	The name of the machine.
SID	The process ID and the instance ID (after the colon).
CPU	The amount of time used by the process in 1/100th of a second.
Memory	The process heap memory used in kilobytes (KB).

Table 8–9 Monitor status connection information

Column	Description
Current	The number of current in-bound connections to the process.
Last Time	The time of the last in-bound connection to the process in the format [(days)hour:minutes:seconds]. This interval is measured since the process started.
Total	The cumulative total number of in-bound connections.

Renaming Applications

Using the Oracle Application Server Manager, you can rename applications. To rename applications, perform the following steps:


1. In the navigational tree, expand Applications by selecting the “+” next to Applications in the tree. This displays a list of all the applications.
2. Expand the application of interest, for example, the Server Status Monitor Application.
3. Select the Configuration folder for the selected application. This shows the application information as in [Figure 8–15](#).
4. Select the Display Name. This brings up the display name form that allows you to change the application’s name.

Reloading Applications

When you add or remove applications, you have to reload the application server for the new configuration to take effect.

Reloading the Application Server


To reload the application server:

1. Select Oracle Application Server in the Oracle Application Server Manager. This displays a list of Oracle Application Server Processes in the right frame.
2. Select “ALL”.
3. Click  to reload the application server.

Reloading Cartridge Servers for All Applications

Using the Oracle Application Server Manager, you can reload cartridge servers for all applications.


To reload all cartridge servers for all applications:

1. In the navigational tree, select Applications. This displays a list of all cartridge server processes.
2. Select the button next to “ALL”.
3. Click  to reload the cartridge servers.

Reloading Cartridge Servers for a Specific Application

Using the Oracle Application Server Manager, you can reload cartridge servers for a specific application.

To reload all cartridge servers for a specific application:


1. In the navigational tree, select the application that you want to stop. This displays a list of cartridges in the application.
2. Select the button next to “ALL”.
3. Click  to reload the cartridge servers.

Deleting Applications

Using the Oracle Application Server Manager, you can delete cartridge servers for all applications or cartridge servers for a specific application.


Deleting Cartridge Servers for All Applications

To delete all cartridge servers for all applications:

1. In the navigational tree, select Applications. This displays a list of all cartridge server processes.
2. Select the button next to “ALL”.
3. Click  to delete the cartridge servers.

Deleting Cartridge Servers for a Specific Application


To delete all cartridge servers for a specific application:

1. In the navigational tree, select the application that you want to delete. This displays a list of cartridges in the application.
2. Select the button next to “ALL”.
3. Click  to delete the selected cartridge servers.

Stopping Processes and Applications

If you make modifications that cannot be reloaded, you have to stop and restart the application server.

To stop all Oracle Application Server processes and listeners:



1. Click on “website40 Site” in the navigational tree. This displays a list hosts.
2. Select the button next to “ALL”.
3. Click  to stop the processes and the listeners.

Stopping Oracle Application Server

To stop the Oracle Application Server:

Note: Before you stop the Oracle Application Server you have to stop the HTTP listeners. Refer to on page 8-32 for details on stopping the listeners.

To stop and restart Oracle Application Server components, perform the following steps:


1. Click on “Oracle Application Server” in the navigational tree to display all the processes.
2. Select the button next to “ALL”.
3. Click  to stop the processes.
4. Select the button next to “ALL”.
5. Click  to start the processes.

Stopping Applications

Using the Oracle Application Server Manager you can stop individual applications (at runtime applications are controlled by cartridge servers). The Application Server Manager allows you to stop cartridge servers for all applications, for a specific application, or to stop just a single cartridge server process.


Stopping Cartridge Servers for All Applications

To stop all cartridge servers for all applications:


1. In the navigational tree, select Applications. This displays a list of all cartridge server processes.
2. Select the button next to “ALL”.
3. Click  to stop the cartridge servers.

Stopping All Cartridge Servers for a Specific Application

To stop all cartridge servers for a specific application:



1. In the navigational tree, select the application that you want to stop. This displays a list of cartridges in the application.
2. Select the button next to “ALL”.
3. Click  to stop the cartridge servers.

Stopping a Single Cartridge Server

1. In the navigational tree, select the application where the cartridge server is running. This displays a list of all cartridge server processes for the application.
2. Select the button next to the cartridge server process you want to stop.
3. Click  to stop the cartridge server.

Stopping and Restarting Listeners

To stop and restart the listeners:

1. Click HTTP Listeners in the navigational tree to display the list of listeners.
2. Select the listeners to stop.
3. Click  to stop the listeners.
4. Select the button next to “ALL”.
5. Click  to start the listeners.

Cartridge and Component Administration

This chapter describes the basic operations you need to perform to add, configure, and manage cartridges and CORBA components using the Oracle Application Server Manager.

Contents

- [Concepts](#)
- [Configuration Roadmap](#)
- [Adding Cartridges](#)
- [Adding CORBA Components](#)
- [Configuring Cartridges](#)
- [Configuring ECO/Java and EJB Components](#)
- [Managing Cartridges](#)
- [Advanced Configuration Topics](#)

Concepts

Applications, cartridges, and CORBA components are the main object types that you use when you build and manage Oracle Application Server applications.

The form that your application logic takes and the procedures you use to configure applications depends on the type of your application. Table 9–1 shows the types of application logic that Oracle Application Server supports.

Table 9–1 Location of application logic

If you use:	then your application logic is in:
C cartridge	Shared library (on UNIX) or DLL (on NT)
C++ cartridge	CORBA objects
Enterprise JavaBeans (EJB)	Enterprise JavaBeans
ECO/Java	CORBA objects
JServlet cartridge	Java class files
JWeb cartridge	Java class files
LiveHTML cartridge	embedded scripts in HTML files
ODBC cartridge	Procedures in ODBC databases such as Sybase or Informix
Perl cartridge	Perl scripts
PL/SQL cartridge	Stored procedures in databases

Cartridges

A cartridge consists of code that executes application logic, and configuration data that enables the cartridge to locate and to control certain parameters. For example, the PL/SQL cartridge contains code that enables it to connect to Oracle databases and execute PL/SQL stored procedures in the database. Cartridge-level configuration data contains information such as which Oracle database to connect to and the error level for error reporting.

Cartridges can provide runtime environments for several programming languages. For example, the Perl cartridge contains a Perl interpreter for running Perl scripts.

CORBA Components

CORBA components include ECO/Java and EJB applications written in Java. The cartridge level interface and the configuration parameters for CORBA components are handled differently from other Oracle Application Server cartridges to support IIOP transport for these components. Using CORBA components, the Oracle Application Server Manager provides different configuration options.

For details on the design of CORBA applications, refer to the Oracle Application Server *Developer's Guide: ECO/Java and EJB Applications* and the Oracle Application Server *Developer's Guide: C++ CORBA Applications*.

Configuration Roadmap

Once your web site nodes and HTTP listeners are configured, and you have developed applications to install on your Oracle Application Server, you can add and configure the applications and the cartridges or components within the applications. The following gives you a checklist for adding and configuring applications and cartridges or components.

- Configure the access descriptor to any databases that your application uses. See [Chapter 6, “Database Access Descriptors \(DADs\)”](#) for directions on configuring a database access descriptor (DAD).
- Add the application.
- Configure the application. Application configuration is consistent across all cartridges or application instances.
- Add and configure the application’s cartridges or CORBA components.
 - a. Add and Configure the applications cartridges. For applications of type PL/SQL, JServlet, JWeb, LiveHTML, Perl, or CWeb, you need to add and configure cartridges. See the sections [“Adding Cartridges” on page 4](#) and [“Configuring Cartridges” on page 16](#) for directions on adding and configuring cartridges.

Cartridge configuration sets general parameters, such as cartridge name, display name, tuning parameters and the virtual path for all of a cartridge’s cartridge instances. In addition, each cartridge type has distinct configuration requirements, as listed in the following table:

Application Type	Cartridge Configuration Requirements
PL/SQL	DAD name for destination database
JServlet	Default
JWeb	Default and Java environment variables
LiveHTML	LiveHTML parameter form
Perl	Perl parameter form
CWeb	Cartridge entry point, Cartridge parameter form, CWeb parameter form.

- b. Configure the components for ECO/Java or EJB applications. Configure these applications using the following forms:

- * Application Instance form
- * Object level form

See [“Configuring ECO/Java and EJB Components” on page 30](#) in for directions on configuring your EJB or ECO/Java application instances.

- Configure the ORB. All applications use the ORB internally for communication between the client browser and the Oracle Application Server. You can change the defaults for the ORB and for the security used by the ORB. See [Chapter 5, “Object Request Broker Administration”](#) for directions on configuring ORB parameters.
- Configure Transactions. If your application is transactional, you need to configure your database to be transactional. See [Chapter 11, “Enabling Transactions”](#) for directions.
- Configure Security. If your application requires encryption, authentication or other security features, you need to configure security. See the Oracle Application Server *Security Guide* for details.
- Configure Logging. If you desire logging for your application, see [Chapter 10, “Logging and Database Utilities”](#) for directions on how to enable logging.

Adding Cartridges

An application consists of one or more cartridges which contain the application logic. Adding cartridges to the Oracle Application Server requires that you perform the procedures described below.

- Add the application according to the instructions in [Chapter 8, “Application Administration”](#).
- Configure the application according to the instructions in [Chapter 8, “Application Administration”](#).
- Add one or more cartridges for the application according to the instructions in [“Configuring Cartridges” on page 16](#).
- Configure the cartridges according to the instructions in [“Configuring Cartridges” on page 16](#).

The Add Cartridge form allows you to add a cartridge (see [Figure 9-1](#)). After you add a cartridge the Oracle Application Server Manager displays additional forms

where you supply cartridge configuration information. This section shows forms for adding cartridges of all the available cartridge types.

Note: The forms for adding a cartridge manually are different depending on the cartridge type you are adding.

Adding a Cartridge While Adding an Application

If you add an application using the procedures shown in “[Adding Applications](#)” on page 8 in [Chapter 8, “Application Administration](#)” and you select the “Add Cartridge to this Application” button, Oracle Application Server Manager displays the add cartridge form. This allows you to add an application and a cartridge at the same time. The procedures for adding a cartridge in this chapter assume that you previously added an application of the specified type.

Figure 9–1 Add Cartridge form

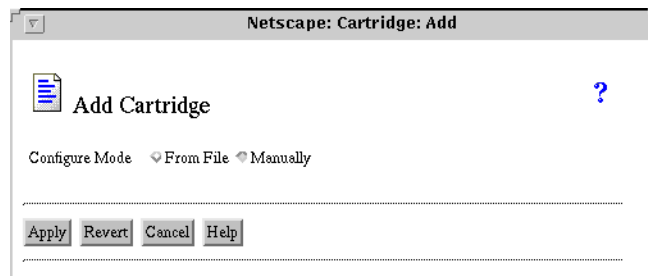



Figure 9–1 shows the initial Add Cartridge form you use to add a cartridge to an application. To display the Add Cartridge form, perform the following steps:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click the “+” next to “Applications” to show the available applications.
4. Click the “/” next to the application that you want to add a cartridge to display the Configuration and Cartridges folders in the left frame.
5. Click on Cartridges to display the Cartridges form in the right frame.

6. On the Cartridges form, in the right frame, click . This brings up the Add Cartridge form shown in [Figure 9-23](#).
7. In the Add Cartridge form select a mode. The options for selecting a mode are:
 - From File — Configure from an existing `<cartridge>.app` file. This takes you to Add Cartridge form, which allows you supply a path for the `.app` file containing configuration information for your cartridge. For more information about `<cartridge>.app` files, see the Oracle Application Server *Cartridge Management Framework* manual.
 - Manually — Configure the cartridge manually.
8. Click Apply.

This displays an Add Cartridge form for the type of application that you are adding. Refer to Table 9-2 for information on the available Add Cartridge forms.

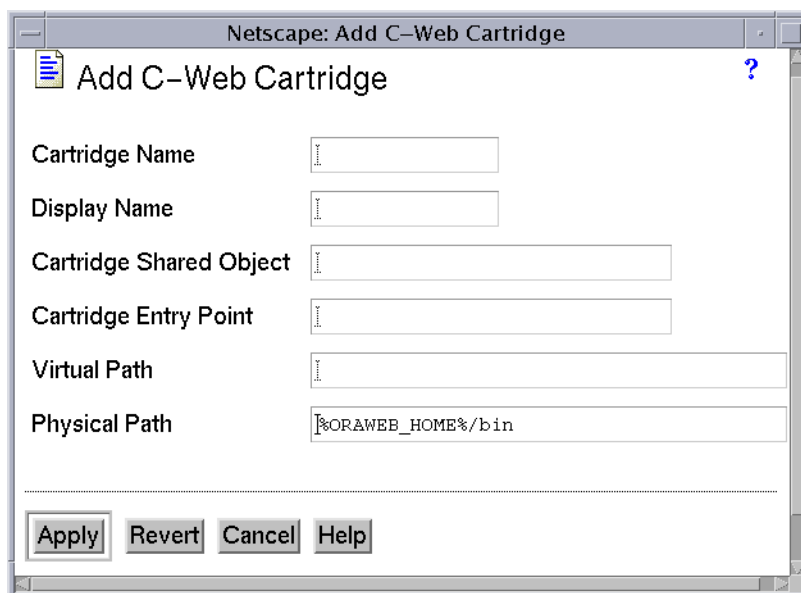
Table 9-2 Add Cartridge Dialogs

Application Type	Refer to Section
CWeb	“Adding a CWeb Cartridge” on page 7
C++ CORBA	“Adding a C++ CORBA Cartridge” on page 8
Enterprise JavaBeans	Chapter 8, “Application Administration” “Adding an EJB or ECO/Java Application”
ECO/Java	Chapter 8, “Application Administration” “Adding an EJB or ECO/Java Application”
JServlet	“Adding a JServlet Cartridge” on page 8
JWeb	“Adding a JWeb Cartridge” on page 9
LiveHTML	“Adding a Cartridge from a Configuration File” on page 15
Perl	“Adding a Cartridge from a Configuration File” on page 15
PL/SQL	“Adding a Cartridge from a Configuration File” on page 15

Adding a CWeb Cartridge

The Add CWeb Cartridge form shows the fields you need to enter to add a CWeb Cartridge (see [Figure 9-2](#))

Figure 9-2 Add CWeb Cartridge



Netscape: Add C-Web Cartridge

Add C-Web Cartridge ?

Cartridge Name

Display Name

Cartridge Shared Object

Cartridge Entry Point

Virtual Path

Physical Path

Cartridge Name

Enter the name for the cartridge. For example, myCcartridge. Do not use commas as part of the cartridge name.

Display Name

Enter the display name for the cartridge. This is the name that appears on the navigational applet tree.

Cartridge Shared Object

Enter the full path to the library .so file (UNIX) or .dll file (NT) that contains the application logic for the cartridge.

Cartridge Entry Point

Enter the name of the entry point function in the .so or .dll file.

Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the cartridge. For example, `/myapp/test`.

The default virtual path for the C cartridge is `/<appName>/<cartName>`. Users can access the C cartridge you add by specifying a URL with the default virtual path or the value that you enter in this field.

Physical Path

Do not supply a value for this field. The default physical path is `%ORAWEB_HOME%/bin`. The physical path is not used by Oracle Application Server. The physical path is available so that you can define a custom path for your cartridge. For example, you can store files that your cartridge uses in the physical path.

Adding a C++ CORBA Cartridge

C++ CORBA cartridges are added by adding the C++ CORBA application. Refer to [“Adding a C++ CORBA Application” on page 12 in Chapter 8, “Application Administration”](#) for information on adding a C++ CORBA application.

Adding a JServlet Cartridge

The Add Cartridge form shows the fields you need to enter to add a JServlet Cartridge (see [Figure 9–4](#)).

Figure 9–3 Add JServlet Cartridge

The screenshot shows a standard Java Swing-style dialog box titled "Cartridge: Add". Inside the dialog, the main heading is "Add A Cartridge" next to a document icon and a question mark icon. Below this heading are four text input fields, each with a label to its left: "Cartridge Name", "Display Name", "Virtual Path", and "Physical Path". At the bottom of the dialog, there is a row of four buttons: "Apply", "Revert", "Cancel", and "Help".

Cartridge Name

Enter the name of the cartridge. For example, myJServletCartridge. Do not use commas as part of the cartridge name.

Display Name

Enter the display name of the cartridge. This is the name that appears on the navigational applet tree.

Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the cartridge. This path is mapped to the path to the location of the class files. For example, /myapp/jServletTest.

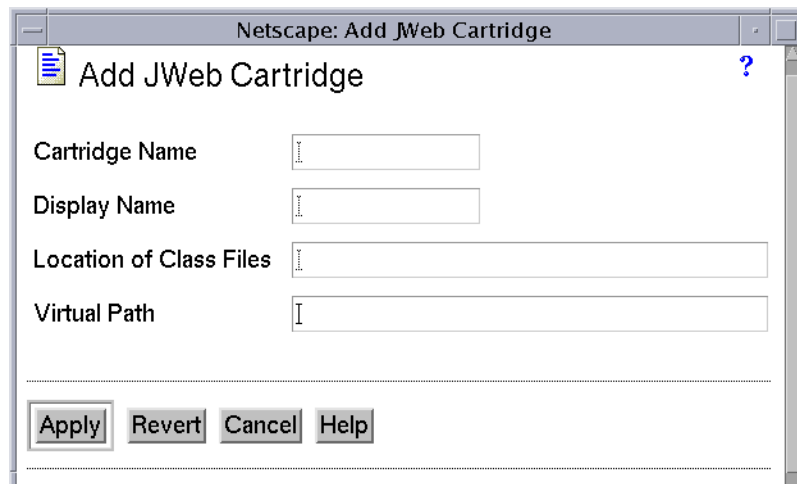
Physical Path

Enter the full path of the directory that contains the cartridge's Java class files. This is in addition to the directories listed in the application's CLASSPATH environment variable. You can leave this field empty if the classes are found in the CLASSPATH.

Adding a JWeb Cartridge

The Add Cartridge form shows the fields you need to enter to add a JWeb Cartridge (see [Figure 9-4](#)).

Figure 9-4 Add JWeb Cartridge



The screenshot shows a Netscape browser window titled "Netscape: Add JWeb Cartridge". Inside the window is a form titled "Add JWeb Cartridge" with a help icon (question mark) in the top right corner. The form contains four text input fields, each with a label to its left: "Cartridge Name", "Display Name", "Location of Class Files", and "Virtual Path". At the bottom of the form are four buttons: "Apply", "Revert", "Cancel", and "Help".

Cartridge Name

Enter the name of the cartridge. For example, myJWebcartridge. Do not use commas as part of the cartridge name.

Display Name

Enter the display name of the cartridge. This is the name that appears on the navigational applet tree.

Location of Class Files

Enter the full path of the directory that contains the cartridge's Java class files. This is in addition to the directories listed in the application's CLASSPATH environment variable. You can leave this field empty if the classes are found in the CLASSPATH.

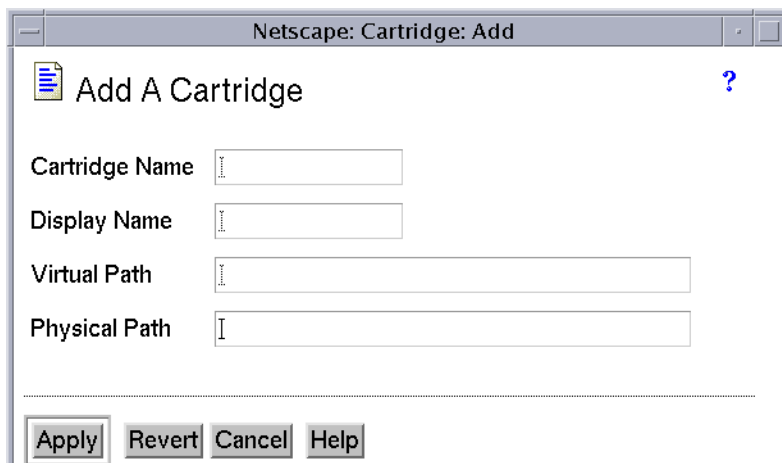
Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the cartridge. This path is mapped to the path to the location of the class files. For example, /myapp/jwebtest.

Adding a LiveHTML Cartridge

The Add Cartridge form shows the fields you need to enter to add a LiveHTML Cartridge (see [Figure 9-5](#)).

Figure 9-5 Add LiveHTML Cartridge



The screenshot shows a Netscape browser window titled "Netscape: Cartridge: Add". Inside the window is a form titled "Add A Cartridge" with a blue question mark icon in the top right corner. The form contains four text input fields, each with a small vertical line icon on the left: "Cartridge Name", "Display Name", "Virtual Path", and "Physical Path". At the bottom of the form are four buttons: "Apply", "Revert", "Cancel", and "Help".

Cartridge Name

Enter the name of the cartridge. For example, myLiveHTMLcartridge. Do not use commas as part of the cartridge name.

Display Name

Enter the display name of the cartridge. This is the name that appears on the navigational applet tree.

Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the LiveHTML cartridge .

Physical Path

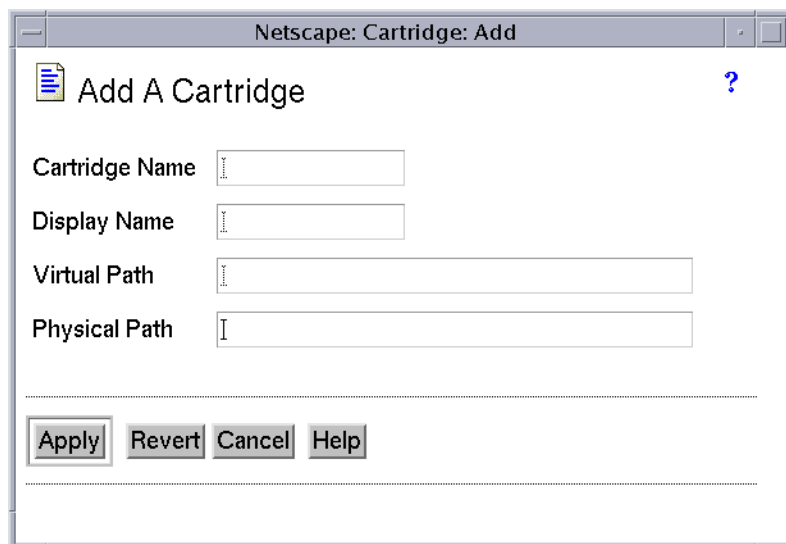
Enter the directory path that leads to files for your LiveHTML cartridge, including files for your LiveHTML application. The virtual path maps to this physical path.

Note: For security reasons, you cannot specify a physical path ending with “..”. But you can use “..” in the physical path setting to indicate an upper directory level. For example, “/routines/../../lib/”.

Adding a Perl Cartridge

The Add Cartridge form shows the fields you need to enter to add a Perl Cartridge (see [Figure 9-6](#)).

Figure 9-6 Add Perl Cartridge



The screenshot shows a Netscape browser window titled "Netscape: Cartridge: Add". The main content area is titled "Add A Cartridge" and contains four text input fields: "Cartridge Name", "Display Name", "Virtual Path", and "Physical Path". Below these fields are four buttons: "Apply", "Revert", "Cancel", and "Help".

Cartridge Name

Enter the name of the cartridge. For example, myPerlcartridge. Do not use commas as part of the cartridge name.

Display Name

Enter the display name of the cartridge. This is the name that appears on the navigational applet tree.

Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the Perl cartridge.

Physical Path

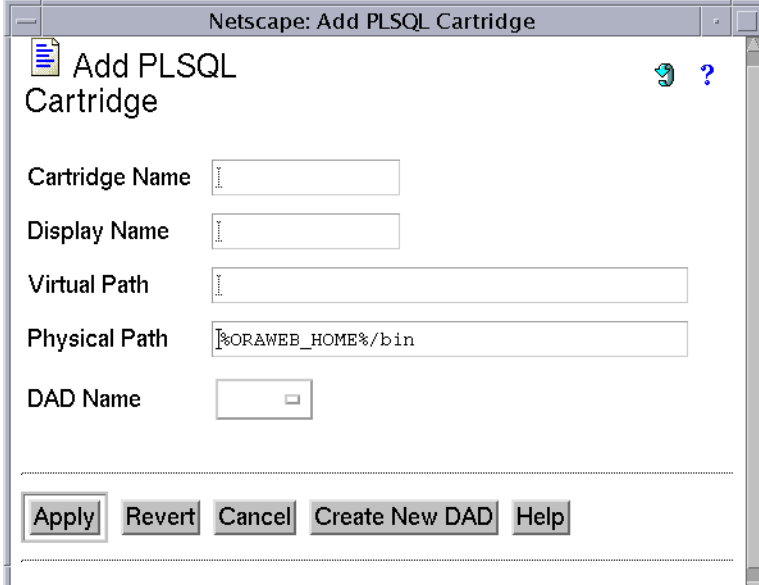
Enter the physical directory path that leads to files for your Perl cartridge, including files for your Perl application. The virtual path maps to this physical path.

Note: For security reasons, you cannot specify a physical path ending with “..”. But you can use “..” in the physical path setting to indicate an upper directory level. For example, “/routines/../../lib/”.

Adding a PL/SQL Cartridge

The Add Cartridge form shows the fields you need to enter to add a PL/SQL Cartridge (see [Figure 9-7](#)).

Figure 9-7 Add PL/SQL Cartridge



Netscape: Add PLSQL Cartridge

Add PLSQL Cartridge

Cartridge Name

Display Name

Virtual Path

Physical Path

DAD Name

Apply Revert Cancel Create New DAD Help

Cartridge Name

Enter the name that the server uses to identify your PL/SQL cartridge.

Display Name

Enter the display name of the cartridge. This is the name that appears on the navigational applet tree.

Virtual Path

Enter a path for the cartridge such that users can specify this path in URLs to invoke the cartridge. The default virtual path for the PL/SQL cartridge is `/<appName>/<cartName>`.

Physical Path

Leave this field as it is if your PL/SQL cartridge is running stored procedures. If your cartridge is running PL/SQL source files (".sql" extension), enter the full path of the directory that contains the PL/SQL source files. Refer to "Executing SQL Files" in the Oracle Application Server *Developer's Guide: PL/SQL and ODBC Cartridges* for details on executing PL/SQL source files.

DAD Name

Select the name of the database access descriptor for this PL/SQL cartridge. If the DAD does not exist create it by clicking the Create New DAD button.

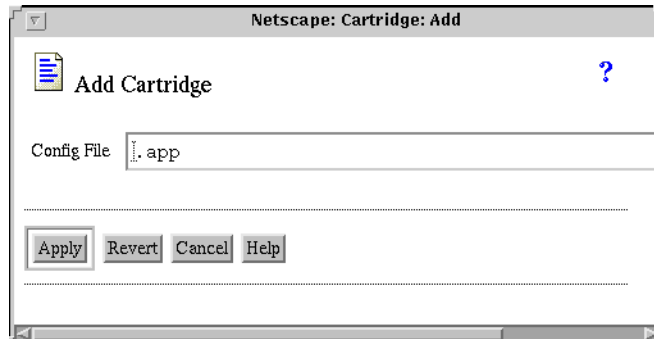
Create New DAD

This displays the Add DAD dialog. A DAD specifies database connection information such as the database to which you want to connect, and the username and password to use to log into the database. When users invoke the PL/SQL cartridge, the cartridge uses the information in the DAD to connect to the database run the stored procedure. Refer to [Chapter 6, "Database Access Descriptors \(DADs\)"](#) for details on creating DADs.

Adding a Cartridge from a Configuration File

The Add Cartridge from configuration file form allows you to add a cartridge and configure it from an existing .app file (see [Figure 9–8](#)).

Figure 9–8 Add Cartridge form (from Configuration File)



Configure File

Enter the file system path to the `<cartridge>.app` file containing the configuration information for your cartridge. For example, `/private/oracle/ows/cartx/plsql/install/cartridge.app`.

For more information about the `<cartridge>.app` file, see the Oracle Application Server *Cartridge Management Framework* manual.

Adding CORBA Components

To add CORBA components, including ECO/Java and EJB components you need to add and configure a CORBA application. For details on adding an ECO/Java or EJB application, refer to [“Adding an EJB or ECO/Java Application” on page 10 in Chapter 8, “Application Administration”](#).

Configuring Cartridges

You can configure cartridge-level parameters using the Configuration forms for each cartridge. The configuration forms vary depending on the cartridge type. If you are in a PL/SQL cartridge, for example, the cartridge forms have different information from the forms for a Perl application.

Two forms are common for most cartridges: the Tuning form which controls tuning and state related parameters and the Virtual Paths form that lets you specify the virtual path and security information for a cartridge.

This sections describes the following configuration tasks:

- [Configuring Tuning Parameters](#)
- [Configuring Virtual Paths](#)
- [Configuring C Cartridge Parameters](#)
- [Configuring JWeb Cartridge Parameters](#)
- [Configuring LiveHTML Cartridge Parameters](#)
- [Configuring ODBC Cartridge Parameters](#)
- [Configuring Perl Cartridge Parameters](#)
- [Configuring PL/SQL Cartridge Parameters](#)

Displaying Cartridge Configuration Forms

To display the configuration forms, expand the applet tree to the cartridge configuration folder by performing the steps below:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display Cartridge configuration forms.
5. Click on the “+” next to the Cartridges folder to display the available cartridges.
6. Click on the “+” next to the cartridge that you want to configure.

7. Click on the “+” next to the Configuration folder to display the configuration forms.
8. Select the configuration form that you want to work with.

The following sections describe how to configure cartridges using the configuration forms.

Configuring Tuning Parameters

The Tuning form controls cartridge tuning parameters. The Tuning form has two options depending on the type of load balancing that is currently in use. [Figure 9–9](#) and [Figure 9–10](#) show the two Tuning forms. For information on setting the Oracle Application Server load balancing options refer to the Oracle Application Server *Performance and Tuning Guide*.

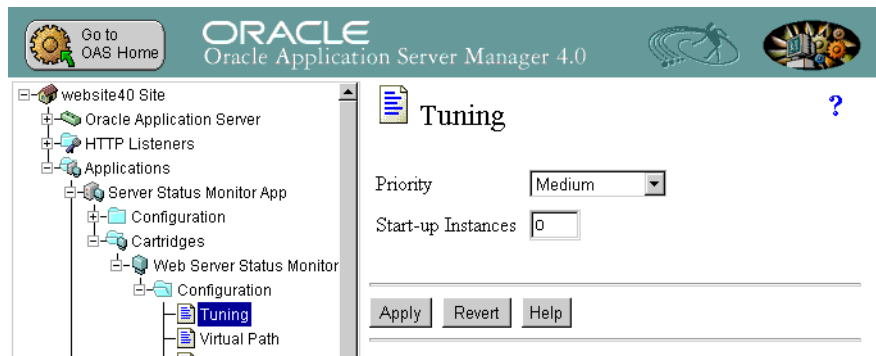
This section covers the following:

- [Cartridge Tuning using Priority load balancing](#)
- [Cartridge Tuning using Min/Max values](#)

Cartridge Tuning using Priority load balancing

In Priority based tuning, you can change the cartridge level priority from the default setting determined at the application level. [Figure 9–9](#) shows the priority based cartridge Tuning form. Refer to “[Configuring Cartridge Server Options](#)” in Chapter 8 for details on setting application level priority load balancing.

Figure 9–9 Cartridge priority Tuning form



Priority Use this field to set the priority level for the cartridge. The cartridge uses this priority level regardless of the priority level set for the application.

Start-up Instances This field defines the number of cartridge instances to start at initial system start-up. These instances remain idle if they are not called before the number of seconds specified in the cartridge timeout (set at the application level). Idle instances are terminated after the cartridge timeout interval, and they are restarted only when called.

Cartridge Tuning using Min/Max values

The Tuning form allows you to define the maximum and minimum number of instances and threads for a cartridge. [Figure 9–10](#) shows the Min/Max Values Tuning form. See the Oracle Application Server *Performance and Tuning Guide* for more information on setting the Min/Max configuration values.

Figure 9–10 Cartridge min/max Tuning form

The screenshot displays the Oracle Application Server Manager 4.0 interface. On the left, a tree view shows the hierarchy: website40 Site > Oracle Application Server > HTTP Listeners > Applications > Server Status Monitor App > Configuration > Cartridges > Web Server Status Monitor > Configuration > Tuning. The 'Tuning' node is selected. The main area on the right is titled 'Tuning' and contains five input fields with the following values: Minimum # of Instances (1), Maximum # of Instances (10), Minimum # of Threads (1), Maximum # of Threads (1), and Maximum # of Clients (for Stateless Cartridges) (0). At the bottom of the form are three buttons: Apply, Revert, and Help.

Minimum # of Instances Specifies the minimum number of cartridge instances that can run within each cartridge server process for the application.

When a cartridge server process starts up, it starts the minimum number of cartridge instances for each cartridge. As it receives requests beyond what the minimum number of instances can handle, it creates more instances, up to the maximum number.

Maximum # of Instances Specifies the maximum number of cartridge instances that can run within each cartridge server process for the application.

Minimum # of Threads Specifies the minimum number of threads available to service requests for all instances of a cartridge.

Because the cartridge server is multi-thread (MT) safe, it can run multiple threads for one or more cartridge instances, but only if these instances are MT-safe. When you configure your cartridges, you specify the number of cartridge instances and the number of threads that a cartridge server can run (see [“Advanced Configuration Topics” on page 41](#) for more information on threads and cartridge instances).

Maximum # of Threads Specifies the maximum number of threads available to service requests for all instances of a cartridge.

Maximum # of Clients (applies only for stateless cartridges) Indicates the number of clients that a stateless cartridge server can instantiate. If Maximum # of Clients is 0, then the Maximum # of Threads is used to determine Maximum # of Clients (see [“Max Clients” on page 9-42](#) for more information on this parameter).

Configuring Virtual Paths

The Virtual Paths form allows you to specify the virtual path for a cartridge and also specify authentication schemes for the virtual path (see [Figure 9-11](#)). See the Oracle Application Server *Security Guide* for authentication details.

Figure 9–11 Virtual Paths form

Virtual Path	Physical Path
/owsapps/owsstat	%ORAWEB_HOME%/bin
/owsstat	%ORAWEB_HOME%/bin
/	/
/	/
/	/

Virtual Path	Scheme	Realm
/		/
/		/
/		/

Virtual Path Enter a path for the cartridge such that users can specify this path in URLs to invoke the cartridge. This path is mapped to the physical path. For example, /myapp/test.

Physical Path Enter the physical directory path that leads to files for your cartridge, including files for your application. The virtual path specified above maps to this physical path. For example, %ORAWEB_HOME%/test.

Protection

The protection parameters allow you to assign authentication schemes to protect individual files and directories used by cartridges.

Virtual Path A virtual filename, directory name, or file pathname.

Use one of the following methods to specify virtual path protection:

- If you want to protect a specific path, then enter the exact path. When a request is received, a check is made to see if that URL matches the entry exactly.
- If you want to protect all sub-paths, then enter a trailing slash.
- If you want to protect all names in a single path (no sub-paths), then enter a trailing asterisk (*).

Examples:

If the virtual path for your cartridge is `/A/B/C`, then:

1. `/A/B/C` means to protect the URL `/A/B/C` only.
2. `/A/B/C/D55` means to protect the URL `/A/B/C/D55` only.
3. `/A/B/C/` means to protect the URL `/A/B/C` and all URLs below (for example., `/A/B/C/D55` and `/A/B/C/E/F`).
4. `/A/B/C3*` means to protect any URL that starts with `/A/B/C3` (like `/A/B/C327`, but not any sub-paths like `/A/B/C/D`).

If there are two or more entries, the last entry specified takes precedence over any previous entries.

In previous releases of the Oracle Application Server, a trailing `/` was not required to protect URLs below a specified URL, as shown in Example 3. If you are migrating from a previous release and want to specify protection for all the URLs below a virtual path, be sure to include the trailing slash (for example, use `/plsqli_dbaplsql/` instead of `/plsqli_dbaplsql`).

Scheme A pull-down menu that lets you select the authentication scheme. If you use this field to specify an authentication scheme, you must specify a realm in the Realm field.

- Basic
- Digest
- Basic_Oracle (database)
- Crypt
- Certificate

Realm Specifies one of the realms defined in the Basic, Digest, or Basic_Oracle forms.

&/| A pull-down menu to assign more than one authentication scheme to the specified files or directories.

If you choose '&', requesters are required to satisfy all configured schemes. If you choose '|', requesters may access the specified files or directories if they satisfy at least one of the two schemes.

IP/Domain A pull-down menu that specifies whether to apply IP-based or domain-based authentication, or neither to the specified files or directories. If you use this field, you must specify the Group field.

Group Specifies one of the groups defined in the IP or Domain forms.

Configuring C Cartridge Parameters

For C cartridges, the cartridge configuration folder contains four forms: the Tuning form, the Virtual Paths form, the CWeb Parameters form, and the Cartridge Parameters form.

Tuning Form

The Tuning form controls performance tuning and cartridge state related parameters. See ["Configuring Tuning Parameters" on page 9-17](#) for a description of the Tuning form.

Virtual Paths Form

The Virtual Paths form enables you to specify the virtual path for the cartridge and also any authentication schemes for the virtual path. See the Oracle Application Server *Security Guide* for authentication details. See ["Configuring Virtual Paths" on page 9-19](#) for a description of the Virtual Paths form.

CWeb Parameters Form

The CWeb Parameters form lets you specify CWeb options. Table 9-3 shows the CWeb parameters that you can set using the CWeb Parameters form.

Table 9-3 CWeb Parameters

Name	Value
Cartridge Shared Object	Sets the shared object file or DLL file containing the code for the cartridge
Cartridge Entry Point	The entry-point function in the shared object or DLL.

Table 9–3 (Cont.) CWeb Parameters

Name	Value
Stateless	<p>Determines whether the cartridge is stateless. A C cartridge can be stateless or stateful. A stateless cartridge can run multiple threads per cartridge instance, while a stateful cartridge can run only one thread per cartridge instance. A stateless cartridge can also use the max clients parameter to set the maximum number of clients that the cartridge can serve at the same time. See "Configuring Tuning Parameters" on page 9-17 for details on setting the max clients parameter</p> <p>Set a cartridge to stateless only if the cartridge is thread-safe. See "Stateful Mode" on page 41 for more information on stateless cartridges</p>

Figure 9–12 CWeb Parameters form

The screenshot displays the 'CWEB Parameters' configuration window. On the left, a tree view shows the navigation path: website40 Site > Oracle Application Server > HTTP Listeners > Applications > Server Status Monitor App > Cartridges > Web Server Status Monitor > Configuration > CWEB Parameters. The main area on the right contains the following settings:

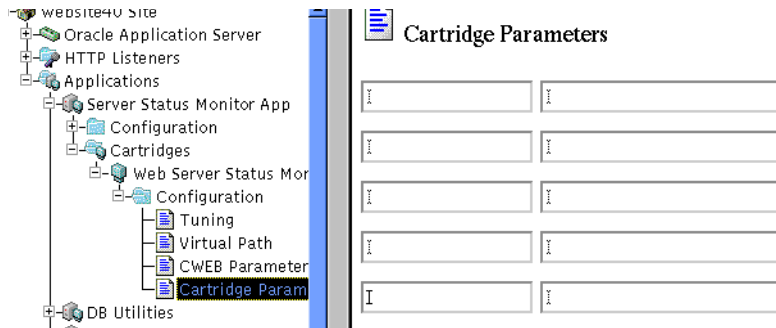
- Cartridge Shared Object:** A text field containing the path `%ORACLE_HOME%\bin\wstat.dll`.
- Cartridge Entry Point:** A text field containing the value `testentry`.
- Stateless:** A checkbox labeled 'False' which is currently unchecked.

At the bottom of the form are four buttons: 'Apply', 'Revert', 'Cancel', and 'Help'.

Cartridge Parameters Form

The Cartridge Parameters form lets you define name-value pairs of parameters for a C cartridge to use.

Figure 9–13 Cartridge Parameters form



Configuring JWeb Cartridge Parameters

For JWeb cartridges, the cartridge configuration folder contains two forms: the Tuning form and the Virtual Paths form.

Tuning Form

The Tuning form controls performance tuning and cartridge state related parameters. See ["Configuring Tuning Parameters" on page 9-17](#) for a description of the Tuning form.

Virtual Paths Form

The Virtual Paths form lets you specify the virtual path for the cartridge and also any authentication schemes. See the Oracle Application Server *Security Guide* for authentication details. See ["Configuring Virtual Paths" on page 9-19](#) for a description of the Virtual Paths form.

Configuring LiveHTML Cartridge Parameters

For LiveHTML cartridges, the cartridge configuration section contains two forms: the Virtual Paths form and the LiveHTML Parameters form.

Virtual Paths Form

The Virtual Paths form lets you specify the virtual path for the cartridge and also any authentication schemes for the virtual path. See the Oracle Application Server *Security Guide* for authentication details. See ["Configuring Virtual Paths" on page 9-19](#) for a description of the Virtual Paths form.

LiveHTML Parameters Form

The LiveHTML Parameters form lets you configure parameters specific to LiveHTML cartridges (see [Figure 9–14](#)). Table 9–4 shows the LiveHTML parameters that you can set in the LiveHTML Parameters form.

Figure 9–14 *LiveHTML Cartridge Configuration form*

The screenshot shows the Oracle Application Server Manager 4.0 interface. The left-hand tree view includes nodes for 'website40 Site', 'Oracle Application Server', 'HTTP Listeners', 'Applications', 'Server Status Monitor App', 'DB Utilities', 'jservlet', 'jweb', 'livehtml', 'Configuration', 'Cartridges', and 'cart'. Under 'Configuration', there are sub-nodes for 'Tuning', 'Virtual Path', and 'LiveHTML Parameters', which is currently selected. The main content area is titled 'LiveHTML Cartridge Configuration' and contains the following settings:

- Enable LiveHTML?**: ☒
- Parse LiveHTML Extensions Only?**: ☒
- LiveHTML Extensions**:
- Enable Exec Tag?**: ☒
- Enable ICX Tag?**: ☒
- Check for <BODY> tag in ICX?**: ☒
- Default page**:
- Enable Script Execution?**: ☒
- Script Page Extension**:
- Default Scripting Language**:
- MaxRequests**:
- Perl Script Configuration**:
 - Perl Application Library Paths**:

At the bottom of the form are three buttons: 'Apply', 'Revert', and 'Help'.

Table 9–4 *LiveHTML Parameters*

Option	Description	Default
Enable LiveHTML	Determines whether the LiveHTML cartridge is enabled. If not enabled, SSI commands, scripting commands, and Web Application Objects are not interpreted. If you want to enable only some cartridge features, you can enable the LiveHTML cartridge, and disable the features that you do not want.	Enabled
Parse LiveHTML Extensions Only	Determines whether the cartridge should parse files with the extensions specified in the “LiveHTML Extensions” field. If enabled, the cartridge parses files with extensions listed in the “LiveHTML Extensions” field only. If not enabled, the cartridge parses all files regardless of extension.	Enabled

Table 9–4 (Cont.) LiveHTML Parameters

Option	Description	Default
LiveHTML Extensions	<p>The list of file extensions handled by the cartridge. This field is used only if you have enabled the “Parse LiveHTML Extensions Only” field.</p> <p>You can configure the cartridge to process all HTML files, that is, set the extension list to include “html”. However, unless all your HTML files actually use SSI, this degrades performance.</p>	html shtml lhtml
Enable Exec Tag	Determines whether the <code>exec</code> SSI command is interpreted by the cartridge.	Enabled
Enable ICX Tag	Determines whether the <code>request</code> command is interpreted by the cartridge.	Enabled
Check for <BODY> tag in ICX	<p>Determines whether the cartridge checks for the <BODY> tag inside the response to an ICX request. (ICX requests are sent using the <code>request</code> command.)</p> <p>If enabled, only data in the <BODY> section of the ICX response is included in the page that sent the <code>request</code> command. If no <BODY> section is found in the ICX response, the cartridge raises an error.</p> <p>If not enabled, the entire ICX response is included in the page.</p>	Enabled
Default page	The page returned to the client if the URL does not specify a file.	index.html
Enable Script Execution	Determines whether embedded scripts in the files are interpreted by the cartridge.	Enabled
Script Page Extension	The list of file extensions that the cartridge checks for embedded scripts.	hsp hsa asp asa
Default Scripting Language	The default scripting language. Currently, Perl is the only language supported. You can specify a different language for the scope of a page or script block. See “Specifying Scripting Languages” in the Oracle Application Server <i>Developer’s Guide: LiveHTML and Perl Applications</i> .	Perl
Max Requests	<p>The number of requests that a cartridge server handles before it terminates.</p> <p>This field can be useful while you are developing LiveHTML applications. If your page calls a Perl library, the Perl interpreter caches the Perl library and uses the cached version for subsequent requests. If you modify the library, you want the interpreter to load the new version. To do this, you have to terminate the cartridge server process so that a new cartridge server process (with a new Perl interpreter) would handle the request. A quick way of doing this is to set the Max Requests value to 1.</p>	none the cartridge server can handle an unlimited number of requests
Perl Application Library Paths	<p>The directories that the Perl interpreter searches in for Perl libraries.</p> <p>If you add paths to this option, you should use full pathnames. If specifying multiple directories, use “/” to delimit each directory.</p>	the current working directory of the cartridge server process

Configuring ODBC Cartridge Parameters

The Oracle Application Server Manager does not include forms for configuring ODBC cartridges. Configure ODBC cartridges using the manual procedures found in Chapter 10 of the Oracle Application Server *Developer's Guide: PL/SQL and ODBC Applications*.

Configuring Perl Cartridge Parameters

For Perl cartridges, the cartridge configuration section contains two forms: the Virtual Paths form and the Perl Parameters form.

Virtual Paths Form

The Virtual Paths form enables you to specify the virtual path for the cartridge and also any authentication schemes for the virtual path. See the Oracle Application Server *Security Guide* for authentication details. See ["Configuring Virtual Paths" on page 9-19](#) for a description of the Virtual Paths form.

Perl Parameters Form

The Perl Parameters form lets you define parameters specific to the Perl cartridge (see [Figure 9-15](#)). Table 9-5 shows the available Perl parameters.

Figure 9-15 Perl Cartridge Configuration form

The screenshot displays the Oracle Application Server Manager interface. On the left, a tree view shows the hierarchy: website40 Site > Oracle Application Server > HTTP Listeners > Applications > Perl App > Configuration > Cartridges > cart1 > Configuration > Perl Parameters. The 'Perl Parameters' item is selected. On the right, the 'Perl Cartridge Configuration' form is shown. It contains the following fields:

ARCHLIB	%ORAWEB_HOME%../cartx/perl/lib/sun4-solaris/5.003
PRIVLIB	%ORAWEB_HOME%../cartx/perl/lib
SITEARCH	%ORAWEB_HOME%../cartx/perl/lib/site_perl/sun4-solaris
SITELIB	%ORAWEB_HOME%../cartx/perl/lib/site_perl
Initialization Script	%ORAWEB_HOME%../cartx/perl/lib/perlinit.pl
Max Requests	10

Table 9–5 Perl parameters

Name	Value	Default
ARCHLIB	The path for architecture-dependent libraries.	%ORAWEB_HOME%/../cartx/perl/lib/sun4-solaris/5.003
PRIVLIB	The path for private libraries.	%ORAWEB_HOME%/../cartx/perl/lib
SITEARCH	The path for site-specific architecture-dependent libraries.	%ORAWEB_HOME%/../cartx/perl/lib/site_perl/sun4-solaris
SITELIB	The path for site-specific libraries.	ORAWEB_HOME%/../cartx/perl/lib/site_perl
Initialization Script	The script that is run when an instance of the Perl cartridge starts up.	%ORAWEB_HOME%/../cartx/perl/lib/perlinit.pl
Max Requests	<p>The number of requests that a cartridge server handles before it terminates.</p> <p>This field can be useful while you are developing Perl applications. If you call a Perl library, the Perl interpreter caches the Perl library and uses the cached version for subsequent requests. If you modify the library, you want the interpreter to load the new version. To do this, you have to terminate the cartridge server process so that a new cartridge server process (with a new Perl interpreter) would handle the request. A quick way of doing this is to set the Max Requests value to 1.</p>	There is no default, which means that the cartridge server can handle an unlimited number of requests.

Configuring PL/SQL Cartridge Parameters

For PL/SQL cartridges, the cartridge configuration folder contains three forms: the Tuning Form, the PLSQL Parameters form, and the Virtual Paths form.

Tuning Form

The Tuning form controls performance tuning and cartridge state related parameters. See ["Configuring Tuning Parameters" on page 9-17](#) for a description of the Tuning form.

Virtual Paths Form

The Virtual Paths form lets you specify the virtual path for the cartridge and also any authentication schemes for the virtual path. See the Oracle Application Server *Security Guide* for authentication details. See ["Configuring Virtual Paths" on page 9-19](#) for a description of the Virtual Paths form.

PL/SQL Parameters Form

The PL/SQL Parameters form lets you specify PL/SQL specific parameters (see [Figure 9-16](#)). The information you can configure includes the following:

Figure 9-16 *PL/SQL Parameters form*

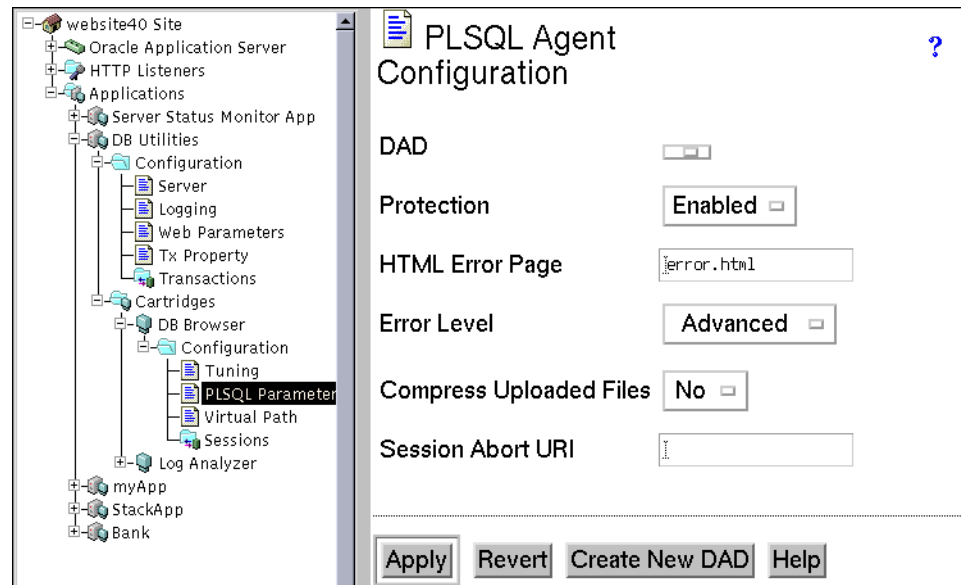


Figure 9-17 *PL/SQL Parameters*

Name	Value	Default
DAD	The DAD associated with the PL/SQL cartridge.	
Protection	Whether protection is enabled.	

Figure 9–17 (Cont.) PL/SQL Parameters

Name	Value	Default
HTML Error Page	The HTML page to display if the database running the PL/SQL procedure returns an error. Note: Errors returned by the PL/SQL cartridge, will not cause this page to load. Instead the application level error page will load. The application level error page is set in the Web Parameters form.	
Error Level	How much error information to display.	
Compress Uploaded Files	Whether uploaded files are stored in compressed format.	
Session Abort URI		

Configuring ECO/Java and EJB Components

You can configure ECO/Java and EJB applications generally, using the application level configuration forms, or by application instance for a number of objects. Objects also have configuration forms for setting Object level parameters. This section describes the following configuration tasks:

- [Configuring Application Instances](#)
- [Configuring CORBA Objects](#)

Configuring Application Instances

To display the Application Instance configuration forms, expand the applet tree to the Application Instance folder by performing the steps below:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display configuration forms.
5. Click on the “+” next to the “Application Instances” folder to display the available tuning forms.
6. Select the configuration form that you want to work with.

The following sections describe how to configure Application Instance and Objects using the configuration forms.

Configuring Tuning Parameters (for Application Instances)

The Application Instance Tuning form controls CORBA application instance tuning parameters (see [Figure 9-18](#)). The Tuning form allows you to define the maximum and minimum number of instances and threads for all objects running within an application instance. See the Oracle Application Server *Performance and Tuning Guide* for more information on setting the Min/Max configuration values.

Figure 9-18 *Application Instance Tuning form*

Tuning	
Minimum # of Instances	10
Maximum # of Instances	10
Minimum # of Threads	1
Maximum # of Threads	10
Maximum # of Clients (for Stateless Cartridges)	10
<input type="button" value="Apply"/> <input type="button" value="Revert"/> <input type="button" value="Help"/>	

Minimum # of Instances Specifies the minimum number of object instances that can run within the application.

When an application instance server process starts up, it starts the minimum number of instances for each object. As it receives requests beyond what the minimum number of instances can handle, it creates more instances, up to the maximum number.

Maximum # of Instances Specifies the maximum number of object instances that can run within each server process for the application.

Minimum # of Threads Specifies the minimum number of threads available to service requests for all instances of objects.

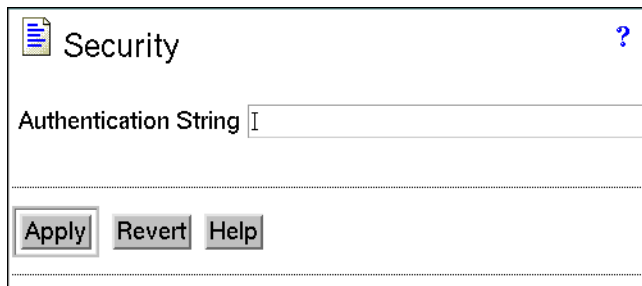
Maximum # of Threads Specifies the maximum number of threads available to service requests for all instances of a cartridge.

Maximum # of Clients (applies only for stateless cartridges) Indicates the number of clients that a stateless cartridge server can instantiate. If Maximum # of Clients is 0, then the Maximum # of Threads is used to determine Maximum # of Clients (see ["Max Clients" on page 9-42](#) for more information on this parameter).

Configuring Security Parameters (for Application Instances)

The Security Configuration form allows you to protect an object. You associate an object with authentication strings, which define the protection. Authentication strings are based on the authentication server component of the application server. The authentication server provides authentication and restriction schemes that you can use to protect cartridges, CGI scripts, and HTML pages (see [Figure 9-19](#)). The Security form allows you to define authentication string.

Figure 9-19 Application Instance Security form



For more information about protecting objects, see the Developer's Guide books. For descriptions of each scheme in the authentication server, see the Oracle Application Server *Security Guide*.

Authentication String To protect an object with an authentication server scheme, you assign an authentication string to the object. The authentication string has the following format:

```
<scheme>(<realm>) [ {"|" | "&"} & <scheme>(<realm>) ... ]
```

An authentication string can consist of more than one “*scheme (realm)*” specification (see [“Configuring Virtual Paths” on page 19](#) for more details on security schemes and realms).

For example, you can have an authentication string that looks like:


```
Basic(myProject) & IP(buildingOne)
```

This authentication string consists of two parts, and a client must fulfill both parts to access the protected object. The client must provide a username/password for the myProject realm and its IP address must be in the buildingOne realm.

You can also connect the parts with the | character. This or operator indicates that a client needs to fulfill only one part of the authentication string. For example, in the following authentication string a client can access the object if it provides a username/password for the myProject realm, or if its IP address is in the buildingOne realm:

```
Basic(myProject) | IP(buildingOne)
```

If an object does not have the authenticationString parameter, it is an unprotected object.

Authentication strings apply only to clients external to the application server. If a client is, for example, another CORBA object or other application server components such as JServlet or LiveHTML cartridge, access restrictions do not apply. You can use the noaccess scheme to define objects that can be accessed only from within the application server

Configuring Instance Parameters (for Application Instances)

The application instances Instance Parameters form allows you to set the Object timeout parameter on the application instance level (see [Figure 9-20](#)).

Figure 9-20 *Application Instances Parameters form*



Bean Parameters ?

Timeout (in seconds)

Apply Revert Help

Configuring CORBA Objects

To display the CORBA configuration forms, expand the applet tree to the Application Instance or Objects folder by performing the steps below:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display configuration forms.
5. Click on the “+” next to the “Objects” folder to display the available Objects.
6. Click on the “+” next to the Object that you want to configure.
7. Select the configuration form that you want to work with.

The following sections describe how to configure Application Instance and Objects using the configuration forms.

Configuring Object Parameters (for Objects)

The object level Object Parameters form allows you to set the object timeout parameter on the Object level (see [Figure 9-21](#)).

Figure 9-21 Application Instances Parameters form



The screenshot shows a web form titled "Bean Parameters" with a blue question mark icon in the top right corner. Below the title, there is a label "Timeout (in seconds)" followed by a text input field containing the number "10". At the bottom of the form, there are three buttons: "Apply", "Revert", and "Help".

Configuring Object Environment (for Objects)

The object level Object Environment form allows you to set object-level parameters. (see [Figure 9-22](#)).

Figure 9–22 Application Instances Parameters form

The screenshot shows a web-based form titled "Java Environment". The form contains the following elements:

- A title bar with the text "Java Environment" and a blue question mark icon on the right.
- A text input field labeled "initialStackSize" with the value "20" entered.
- Five pairs of empty text input fields arranged vertically.
- At the bottom, three buttons: "Apply", "Revert", and "Help".

Managing Cartridges

The Cartridges form allows you to manage cartridges (see [Figure 9–23](#)). To display the Cartridges form, perform the following steps:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display Cartridge information.
5. Click on the cartridges folder. If you select the “+” next to cartridges, you will see a list of the available cartridges, and the information in the right frame does not display the Cartridges form.

Table 9–6 describes the information shown in the Cartridges form.

Figure 9–23 Cartridges form

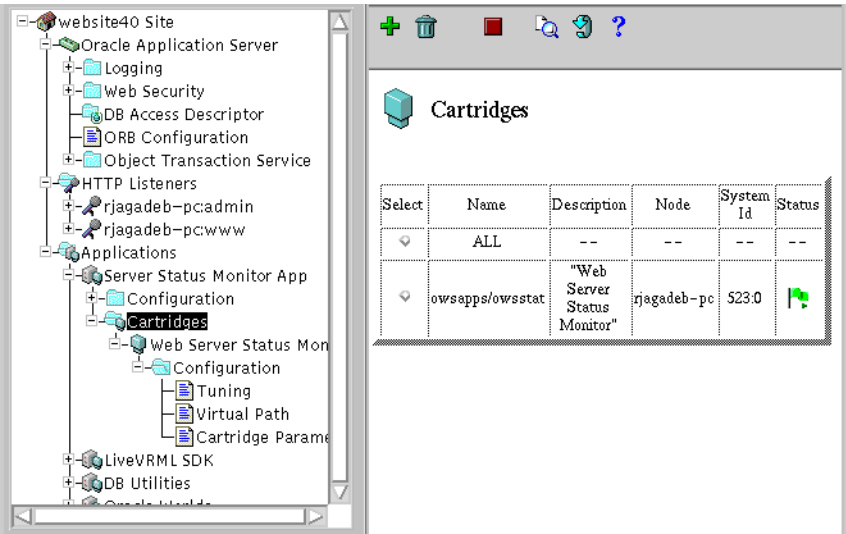







Table 9–6 Cartridges form fields

Operation	Comments
Select	Allows you to select a particular cartridge. Use the option to select a cartridge to perform cartridge operations.
Name	Contains a shortened name for a cartridge. For example, owsapps/owstat.
Description	Contains a descriptive name for an application. This is the name that is shown on the applet tree. For example, Web Server Status Monitor.
Node	Specifies the different machine names. For example, isp-sys1.
System ID	Indicates the cartridge process ID and instance number. For example, 2342 : 1 where 2342 is the process ID and 1 is the instance of the process. If a cartridge is down, it does not display a system ID.
Status	Indicates whether the cartridge is up or down. A red flag indicates the cartridge is down; a green flag indicates the cartridge is up.

Table 9–7, “Cartridge operations” describes the available cartridge operations. The following sections describe these cartridge operations.

Table 9–7 Cartridge operations

Operation	Comments
Add	Allows you to add a new cartridge. You need to supply cartridge information in the Add Cartridge form. Click  in the Cartridge form to add a cartridge.
Delete	Removes a cartridge. A pop-up confirmation window confirms the deletion. Click  in the Cartridge form to delete a Cartridge.
Stop	Stops a cartridge. Click  in the Cartridge form to stop a cartridge.
Monitor	Prints out cartridge statistics. Click  in the Cartridge form to monitor a cartridge.
Update Page	Updates the cartridge information form. Click  to update the cartridge information page.

Monitoring Cartridges

The Cartridges Monitoring form displays status information for the cartridge you select to monitor (see [Figure 9–24](#)). To display the Cartridges Monitoring form, perform the following steps:


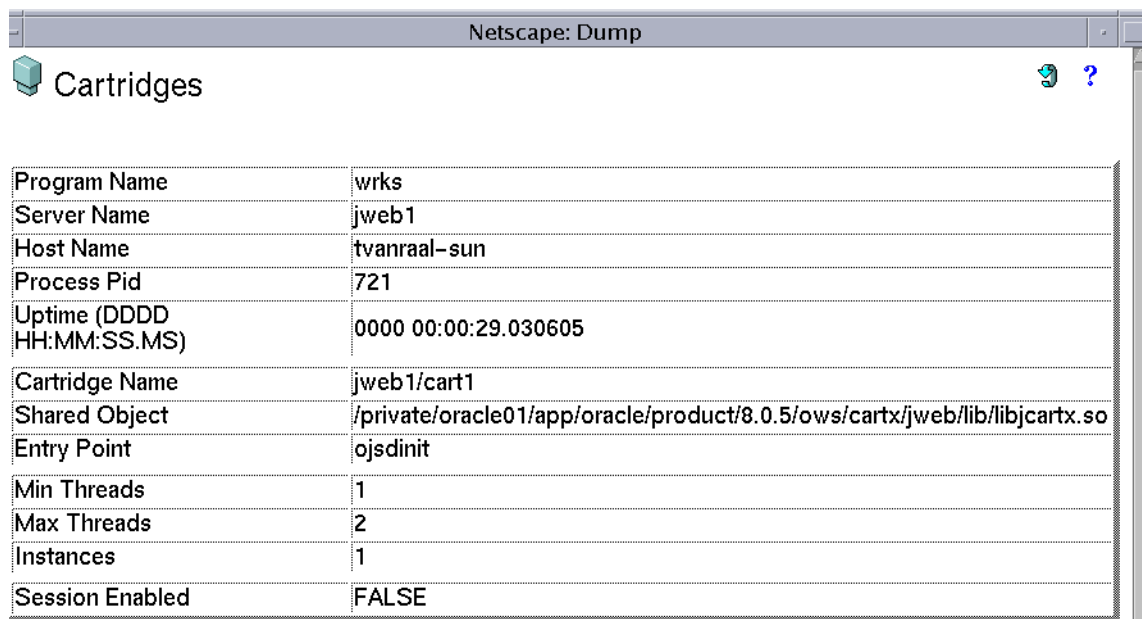
1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display Cartridge information.
5. Click on the “Cartridges” folder.
6. Using the Select field in the Cartridge form, select the cartridge you want to monitor. Click  to monitor the selected cartridge.

Table 9–8 describes the cartridge monitor form fields.

Note: All the monitoring statistics are a rough indication of the state of the server. They are not precise.

Figure 9–24 Cartridges Monitoring form


Program Name	wrks
Server Name	jweb1
Host Name	tvanraal-sun
Process Pid	721
Uptime (DDDD HH:MM:SS.MS)	0000 00:00:29.030605
Cartridge Name	jweb1/cart1
Shared Object	/private/oracle01/app/oracle/product/8.0.5/ows/cartx/jweb/lib/libjcartx.so
Entry Point	ojsdinit
Min Threads	1
Max Threads	2
Instances	1
Session Enabled	FALSE

Table 9–8 Cartridge Monitor fields


Field	Description
Program	The executable that runs the application. It is defined to be wrks .
Server Name	The name of the application.
Host	The host name of the machine.
PID	The process ID.
Up Time	The length of time that the cartridge server has been running displayed in days, hours, minutes, and seconds. (DDDD HH:MM:SS.SSSSSS)
Cartridge Name	The name of the cartridge: appname/cartname .
Shared Object	The platform-specific implementation of the cartridge. On Windows NT, the shared object typically has the extension .dll . On Solaris, the shared object has the extension .so .
Entry Point	The name of the cartridge entry point function.
Min Threads	Specifies the minimum number of threads available to service requests for all instances of a cartridge.

Table 9–8 (Cont.) Cartridge Monitor fields

Field	Description
Max Threads	Specifies the maximum number of threads available to service requests for all instances of a cartridge.
Instances	The number of cartridge instances that are currently running within the cartridge server process for the application.
Session Enabled	This value lets the dispatcher know that the cartridge is session enabled; requests originating from the same client receive the same cartridge instance. This is true as long as the idle time is less than the specified seconds in the Max session idle time field.

Stopping Cartridges


To stop a cartridge using Oracle Application Server manager, perform the following steps:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager. You connect to the Welcome page by accessing the Oracle Application Server Administration port (for example <http://sname:8888>).
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application containing the cartridge you want to stop.
5. Click on the “Cartridges” folder.
6. Using the Select field in the Cartridge form, select the cartridge you want to stop. Click  to stop a the selected cartridge.

Deleting Cartridges

Oracle Application Server allows you to delete a cartridge. To delete a cartridge, perform the following steps:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager. You connect to the Welcome page by accessing the Oracle Application Server Administration port (for example <http://sname:8888>).

2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.
3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display Cartridge information.
5. Click on the cartridges folder. If you select the “+” next to cartridges, you will see a list of the available cartridges, and the information in the right frame does not display the Cartridges form.
6. Using the Select field in the Cartridge form, select the cartridge you want to delete. Click  to delete the selected cartridge. Stateful and Stateless Mode and Multi-threading

The Oracle Application Server supports two modes of operation for cartridges and ECO/Java components: *stateful* and *stateless*. For Oracle Application Server stateful operations, the Resource Manager only provides a cartridge object reference to a single client. An object instance belongs solely to that client until it is relinquished or until it times out. For Oracle Application Server stateless operation, the Resource Manager provides the same cartridge object reference to more than one client. [Figure 9–25](#) shows the Oracle Application Server stateful and stateless modes.

Oracle Application Server Manager allows you to set the stateful and stateless modes as a configuration option; however, when a cartridge or component is implemented, the developer determines whether it is stateful or stateless. The developer needs to tell the administrator the mode, and the cartridge should be configured accordingly. If there is any doubt about the type of a cartridge you are installing, configure the cartridge in stateful mode.

Displaying Cartridge Information and Changing the Cartridge Display Name

To display cartridge information and optionally change the cartridge display name, use the Configuration Information form. Display the Configuration information form, by expanding the applet tree to the cartridge configuration folder using the steps below:

1. Connect to the Welcome page using your web browser and select Oracle Application Server Manager.
2. Click on the “+” next to a site name to display the components on the site (for example the website40 site). In the Oracle Application Server Manager, you should see “Oracle Application Server”, “HTTP Listeners”, and “Applications”.

3. Click on the “+” next to “Applications” to display the available applications.
4. Click on the “+” next to the application for which you want to display the Cartridge information form.
5. Click on the “+” next to the Cartridges folder to display the available cartridges.
6. Click on the “+” next to the cartridge that you want to display information for, or whose name you want to change.
7. Click on the Configuration folder to display the configuration information.
8. If you want to change the cartridge name, click on the Display Name field. This brings up the Display Name form that allows you to change the cartridge display name.

Advanced Configuration Topics

Stateful Mode

A stateful cartridge or component can run only one thread per instance. Stateful cartridges and components do not work in a multi-threaded environment because they contain structures or variables that are not protected. Using stateful cartridges in a multi-threaded environment can lead to deadlock situations where two, or more, clients wait for resources locked by the other client. In stateful mode, a cartridge object reference belongs to a single client until it is relinquished or until it times out.

For stateful cartridges, the recommended configuration is to have the number of threads equal to or less than the number of instances. You should not have more threads than instances because the extra threads are not used and consume the machine's CPU cycles. You might choose to have fewer threads than instances if you are running on a machine with limited resources.

Stateless Mode

A stateless cartridge or component can run multiple threads safely per cartridge instance. A cartridge designer might want to create stateful cartridges or components to enable each instance to handle more than one request simultaneously. This can significantly enhance performance. Each thread handles a single request.

Note: Only the CWeb, JServlet and JWeb cartridges and ECO/Java or EJB support stateless mode. You have to run other cartridges in stateful mode

The application server uses the following algorithm to determine whether a cartridge is stateless:

- If the stateless attribute is set to true, then the cartridge is stateless; this flag is only available for CWeb and JWeb cartridges.
- The tuning values have a role in determining if a cartridge is stateful or stateless. If there is no stateless flag, the application server looks at the number of cartridges and threads. If max threads is set to a value greater than 1 and max instances is set equal to 1, then the cartridge is stateless. Table 9–9 shows the possible values for the Oracle Application Server stateless and stateful condition as determined by the tuning parameters.
- Otherwise, the cartridge is stateful.

Stateless cartridges have a “max clients” parameter, which specifies the maximum number of clients that a cartridge server can serve at any time. The max clients parameter does not apply to stateful cartridges.

Table 9–9 *Tuning values for stateful and stateless modes*

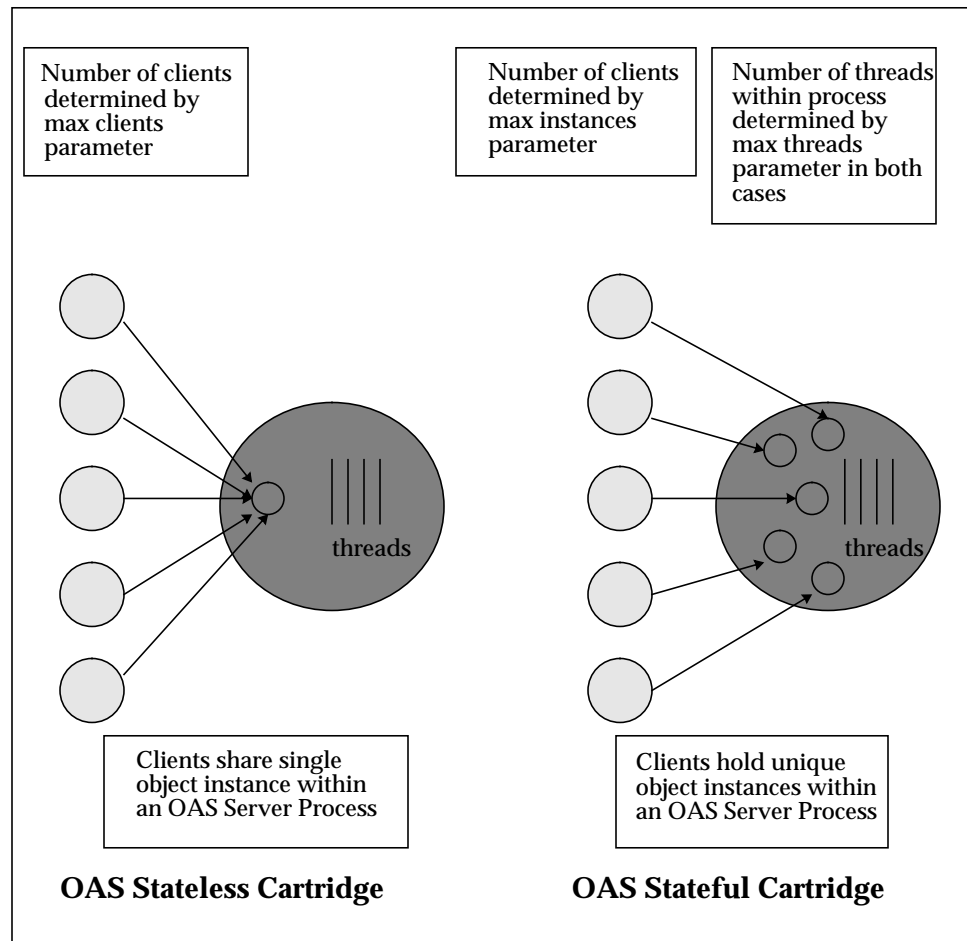
Number of Instances	Number of Threads	Number of Cartridge Types per Server	Classification
1	1	1	
1	1	>1	single threaded stateful
1	>1	any	stateless
>1	any	any	stateful

Max Clients

When a cartridge server process starts up, it starts up the minimum number of cartridge instances for each cartridge. As it receives requests beyond what the minimum number of instances can handle, it starts up more instances, up to the maximum number. Typically, you set the max clients parameter in a stateless cartridge when you have more instances than threads. For example, if you have 5 instances but only 3 threads, you can set max clients to 5 so that the cartridge

server can handle 2 more clients (otherwise, the application server would direct the clients' requests to other cartridge servers). In the cartridge server, if all 3 threads are busy, the cartridge server places the 2 clients' requests in a queue until a thread is available.

Figure 9–25 Stateful and Stateless Modes



Logging and Database Utilities

This chapter discusses the logging features of Oracle Application Server as well as the Database Browser, a utility that allows you to view database information such as users, tables, and views in HTML pages using a web browser.

You can gather information such as how many users are accessing your site, what files are being accessed frequently, and what errors have occurred. Oracle Application Server allows you to configure what information is logged, how the data is formatted, and where the data is stored (filesystem or database). You can also use the Log Analyzer to view, query, or generate reports on the log files.

Contents

- [Configuring Logging Features](#)
- [Installing the Utilities](#)
- [Using the Log Analyzer](#)
- [Using the Database Browser](#)
- [Troubleshooting](#)

Configuring Logging Features

You configure the Logging features and the Log Analyzer through the Oracle Application Server Manager.

The Logger allows you to write error, warning, or other useful messages to a central log repository (a file system or a database). You can then use the Log Analyzer to extract statistics from this data.

The Log Analyzer allows you to monitor Oracle Application Server's activity by displaying statistics such as total bytes transferred or the most frequently accessed

URL, using the raw data stored by the Logger. The information that the Log Analyzer provides you is important in determining how the contents of a web site are being accessed, which links are repeatedly accessed, or who are the most common visitors. This information can help you organize the web site information in a more efficient manner.

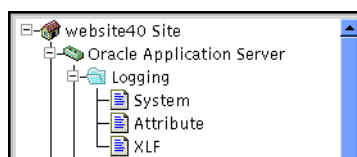
Accessing the Logging Forms

The forms used to configure logging can be located using the following steps:

1. From the Welcome page, select OAS Manager.
2. Expand the Oracle Application Server folder.

The System, Attribute, and XLF logging options are displayed.

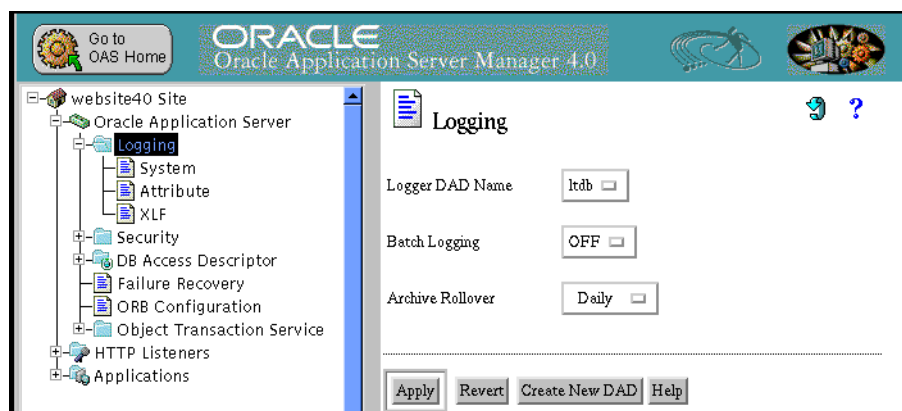
Figure 10–1 OAS Manager Configuration Tree



General Logging Configuration

You can configure general logging information by selecting (without expanding) the Logging folder. The Logging form is displayed:

Figure 10–2 Logging form



This section describes the form's fields:

Logger DAD Name

If you will be logging data to a database, you can select the DAD here. For more information about DADs, see [Chapter 6, “Database Access Descriptors \(DADs\)”](#).

Batch Logging

Specifies whether or not to use batch mode for logging information.

If you use batch logging, the Logger groups the log messages in cached memory until reaching 100 and then stores them as a group. Otherwise, the Logger stores each log message immediately. Possible values are:

- ON - Use batch logging. This is the default.
- OFF - Do not use batch logging.

Archive Rollover

Specifies the time interval when the log file is archived. The records or requests are written to the log file. Archiving the log files is a process of storing or keeping these log files for future references. Log file is archived based on the file size. Possible values are:

- Never - The log file is not archived at a particular time interval.
- Daily - The log file is archived each day at midnight. If no requests are written to the log file at midnight, the log file is archived whenever the next request comes in. For example, if your Archive Rollover value is set to Daily, but no information is written to the log file between 11 p.m. and 3 a.m., the log file is archived at 3 a.m.
- Weekly - The log file is archived each Monday morning at midnight. If no requests are written to the log file all week, the log file is archived whenever the next request comes in.
- Monthly - The log file is archived at midnight on the first day of each month. If no requests are written to the log file all month, the log file is archived whenever the next request comes in.

If Archive Rollover is set to Daily, Weekly, or Monthly, this setting overrides any settings for Archiving File Size (discussed in the section [“Archiving File Size” on page 10-5](#)).

If Archive Rollover is set to Never, Archiving File Size is used to determine when to archive the log file.

Configuring System Logging

The System logging form allows you to configure messages logged by each individual component of Oracle Application Server to the system. These components can include a cartridge, dispatcher, or adapter.

To access the System logging form, expand the Logging folder and select System.

Figure 10–3 System form

The screenshot shows the Oracle Application Server Manager 4.0 interface. On the left, a tree view under 'website40 Site' shows 'Oracle Application Server' expanded, with 'Logging' selected. The 'System' sub-item is highlighted. The main panel displays the 'System' configuration form with the following fields:

Field	Value
Logging	OFF <input type="checkbox"/>
Destination Type	DB <input type="checkbox"/>
Logging Directory	%ORAWEB_ADMIN%/websi
Logging File	web. log
Archiving File Size	10000000
Archiving Directory	%ORAWEB_ADMIN%/websi
Logging Mask	0xFFFFFFFF
Severity Level	15 <input type="checkbox"/>

At the bottom of the form are three buttons: 'Apply', 'Revert', and 'Help'.

This section describes the forms fields:

Logging

Specifies whether to log system messages. Possible values are:

- ON - Log system messages. This is the default.
- OFF - Do not log system messages.

Destination Type

Indicates the type of storage in which to log the information. Possible values are:

- FS - Use a file system as the storage type. This is the default.
- DB - Use a database as the storage type.

If you want to log files to a database, you must enter a DAD in the general Logging form. See [“Logger DAD Name” on page 10-3](#).

When you choose database (DB) as your storage of the log files, you do not have to enter any information that would apply for the filesystem (FS), such as: logging directory or logging file. The log information is stored in the LOGSYSMSG table in the database.

Logging Directory

If you are logging to a filesystem, you can select the directory here.

Note: You cannot substitute a different environment variable for the default, ORAWEB_ADMIN. If you are not using the ORAWEB_HOME, you must give a complete directory path name. For example, /private/OAS/logs/.

Logging File

Specifies the name of the file for logging. The default is **wrb.log**.

Archiving File Size

Specifies the maximum log file size in bytes. Log data is added to a file until it reaches the specified size, a new file is then opened, and the old file is archived in the archive directory. For example, if the maximum log file size is specified as 1000, when the log file reaches 1000 bytes, it is archived and a new log file is created.

The default is 10,000,000 bytes (approximately 10 MB).

Note for Unix users: If you have configured multiple Loggers to write to the same file, archiving occurs when one of the Loggers has written 1000 bytes to the file. In such a scenario, it is possible that the size of the file will be greater than 1000 bytes when it is archived.

Note for Windows users: Multiple Loggers cannot write to the same log file. If you specify the same log file name for more than one Logger, the first Logger writes data to that file. Other Loggers write their data to *logfile_name.PID*, where PID is the process ID of that Logger.

The archived file name is *logfile_name.n*, where *n* is a number from 0 to 999, inclusive. For example, if your log file is called **wrb.log**, archived files are named: **wrb.log.0**, **wrb.log.1**, ..., **wrb.log.999**. After 1000 log files are archived, **wrb.log** will no longer be archived. All messages will continue to be written to **wrb.log**, regardless of the file size. The Logger will not overwrite an existing archived file (unless its size is 0 bytes).

If (in the Logging form) Archive Rollover is set to Daily, Weekly, or Monthly, this setting overrides any settings for Archiving File Size. If Archive Rollover is set to Never, Archiving File Size is used to determine when to archive the log file.

Archiving Directory

Identifies the directory in which to store archived log files. When the log file reaches the archiving file size, it is copied to the archiving directory for storage, and a new log file is created in the logging directory.

Note: You cannot substitute a different environment variable for the default, ORAWEB_ADMIN. If you are not using the ORAWEB_HOME, you must give a complete directory path name. For example, /private/OAS/logs/.

Logging Mask

Specifies a mask to indicate from which component or service within the WRB you want logging messages. The mask values for the components or services are defined as follows:

Table 10–1 Mask values for components/services

Component/Service	Mask Number
All	0xFFFFFFFF
ADP (Adapter)	0x00001FFF

Component/Service	Mask Number
ADP_Spyglass	0x00001001
ADP_Netscape	0x00001002
ADP_IIS	0x00001004
ADP_Netscape_Heartbeat	0x00001008
ADP_Spyglass_Heartbeat	0x0000100a
Dispatcher	0x00002FFF
Dispatcher_Gen	0x00002001
Dispatcher_OSD	0x00002002
Dispatcher_HB	0x00002004
Resource Manager	0x00004FFF
Virtual Path Manager	0x00008FFF
Broker	0x00010FFF
Logger	0x00020FFF
Log Module	0x00040FFF
Auth Module	0x00080FFF
Auth Server	0x00080001
Provider BASIC	0x00080002
Provider DIGEST	0x00080004
Provider IP	0x00080008
Provider DOMAIN	0x00080010
Provider BO	0x00080020
Configuration Provider	0x00100FFF
Proxy Server	0x00200FFF
WRB App Engine	0x00400FFF
WRB Services	0x00800FFF
Content Services	0x00800001

Component/Service	Mask Number
Inter-Cartridge Exchange	0x00800002
Transactional Services	0x00800004
Log Services	0x00800008
Utilities	0x00800010
Java NCX	0x01000FFF
Java Cartridge	0x04000FFF

The masks are used to exclude/include messages that come from a particular component or service. For example, if you take value of resource manager (0x00004FFF) and value of virtual path manager (0x00008FFF), and do an ‘OR’ operation, a mask of (0x0000CFFF) final value is obtained.

Severity Level

Indicates the severity level of system messages logging. For example, if the severity level is 7, then the Logger logs all system messages with a severity level below or equal to 7. You can get more detailed system messages logging, if you use a higher number. The following table describes the severity levels:

Table 10–2 Severity levels

Severity Level	Description	Comments
0	Fatal Errors	A core init, malloc, or any other error that is fatal such that the process has to exit.
1-3	General Errors	An error writing to the file or database or accessing a resource in general.
4-6	Warnings	A missing directory, file, service, or section in a configuration file. These are not fatal as to kill the process.

Severity Level	Description	Comments
7-10	Tracing Information	<ul style="list-style-type: none"> 7- Any messages that occur in initial, shut-down, and reload stages for the cartridges, for example, SSI has been initialized. For the WRB, messages indicating a service has been loaded successfully, terminated successfully, or reloaded. 8- Any messages that occur in authentication and execution stages for the cartridges, for example, request foo has been executed. For the WRB, messages indicating a request for a particular service, or any messages that occurs in the execution path of carrying out the request for a service. 9 - Messages defined by a cartridge or the WRB. 10 - Messages defined by a cartridge or the WRB.
11-15	Debugging Information	Logging of certain variables or arguments, for example, foo=5 .

For information about viewing the system logging entries, see [“System Messages Log Entries” on page 10-33](#).

Configuring Attribute Logging

By default, Oracle Application Server logs information specified in the System or Logging parameters. However, if you would like to log specific information which otherwise would not be logged, you can use the Attributes form. This is also called client-defined attributes.

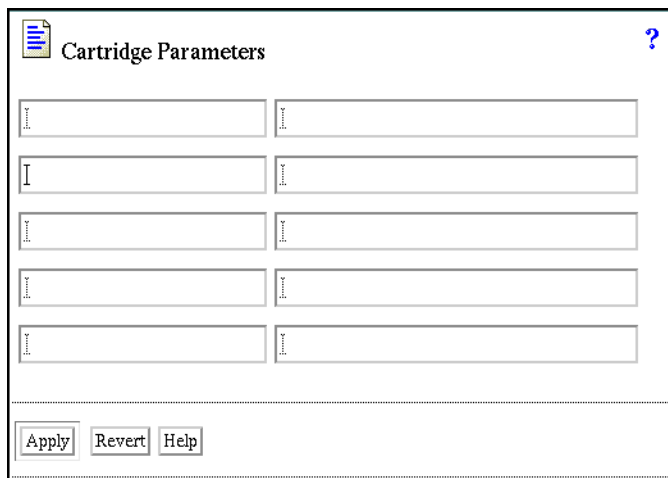
You can use client-defined attributes to monitor a specific component. CPU load and queue length are examples of possible attributes. For example, a cartridge can log specific name-value pairs in the Cartridge Parameters form ([Figure 10-4](#)) which the Logger uses to generate component specific statistics.

For example, `queue_length=10` is a name-value pair. The Logger logs the name of the client-defined attribute and its value.

To access the Cartridge Parameters form:

1. From the Welcome page, select OAS Manager.
2. Expand the site folder.
3. Expand the Applications folder.
4. Expand the Application you wish to configure.
5. Expand the Cartridges folder.
6. Expand the cartridge you wish to configure.
7. Expand the Configuration folder.
8. Click on Cartridge Parameters to display the form:

Figure 10–4 Cartridge Parameters



Cartridge Parameters	
<small>name</small> <input type="text"/>	<small>value</small> <input type="text"/>
<small>name</small> <input type="text"/>	<small>value</small> <input type="text"/>
<small>name</small> <input type="text"/>	<small>value</small> <input type="text"/>
<small>name</small> <input type="text"/>	<small>value</small> <input type="text"/>
<small>name</small> <input type="text"/>	<small>value</small> <input type="text"/>
<input type="button" value="Apply"/> <input type="button" value="Revert"/> <input type="button" value="Help"/>	

Configuring Cartridge Specific Parameters

The Cartridge Parameters form allows you to define specific parameters for a cartridge.

You can define parameters in the form of name-value pairs. Enter the name of each parameter in the left column and its value in the right column. The interpretation of these parameters is entirely determined by your cartridge; they have no meaning to any other Oracle Application Server component.

If you run out of empty lines, click the Apply button. This saves the information you have entered, and gives you five additional blank lines.

[Table 10–3](#) describes client-defined logging attributes:

Table 10–3 Columns in the Logattrib table

Column	Data Type	Description
entry_date	date	The time and date the log entry is recorded.
usec	number	The time (in microseconds) of the log entry.
wrb	varchar2(100)	Indicates a WRB specific value. Currently, it is planned to hold the name of the machine where the WRB resides. This field is used in distributed environments to distinguish between the different WRBs.
component	varchar2(2000)	Indicates the component name such as adapter (ADP), virtual path manager (VPM), or resource manager (RM).
id	number	The process id of the process to be logged.
lname	varchar2(2000)	Name of the client-defined attribute.
lvalue	varchar2(2000)	Value of the client-defined attribute.

To turn on Attribute logging, use the Attribute logging form.

1. From the Welcome page, select OAS Manager.
2. Expand the site folder.
3. Expand the Oracle Application Server folder.
4. Expand the Logging folder and select Attribute to display the form:

Figure 10–5 Attribute form



Logging

Specifies whether or not to use client-defined attribute logging. Possible values are:

- ON - Use client-defined attribute logging. This is the default.
- OFF - Do not use client-defined attribute logging.

Destination Type

Indicates the type of storage in which to log the information. Possible values are:

- FS - Use a file system as the storage type. This is the default.
- DB - Use a database as the storage type.

If you want to log files to a database, you must enter a DAD in the general Logging form. See [“Logger DAD Name” on page 10-3](#).

When you choose database (DB) as your storage of the log files, you do not have to enter any information that would apply for the filesystem (FS), such as: logging directory or logging file. The log information is stored in the LOG-SYMSMSG table in the database.

Logging Directory

Specifies the name of the directory where the log file is located. You cannot change the environment variable which is set by default that is, %ORAWEB_ADMIN%/website40/log. You must give a complete directory path name if you are not using the environment variable. For example, /private/OAS/logs/.

Logging File

Specifies the name of the file for logging. The default is **attrib.log**.

Archiving File Size

Specifies the maximum log file size in bytes. Log data is added to a file until it reaches the specified size; a new file is then opened, and the old file is archived in the archive directory. For example, if the maximum log file size is specified as 1000, when the log file reaches 1000 bytes, it is archived and a new log file is created.

The default is 10,000,000 bytes (approximately 10 MB).

Note for Unix users: If you have configured multiple Loggers to write to the same file, archiving occurs when one of the Loggers has written 1000 bytes to the file. In such a scenario, it is possible that the size of the file is greater than 1000 bytes when it is archived.

Note for Windows users: Multiple Loggers cannot write to the same log file. If you specify the same log file name for more than one Logger, the first Logger writes data to that file. Other Loggers write their data to *logfile_name.PID*, where PID is the process ID of that Logger.

The archived file name is *logfile_name.n*, where *n* is a number from 0 to 999, inclusive. For example, if your log file is called **attrib.log**, archived files are named: **attrib.log.0**, **attrib.log.1**, ..., **attrib.log.999**. After 1000 log files are archived, **attrib.log** will no longer be archived. All messages will continue to be written to **attrib.log**, regardless of the file size. The Logger does not overwrite an existing archived file (unless its size is 0 bytes).

Archiving Directory

Identifies the directory in which to store archived log files. When the log file reaches the archiving file size, it is copied to the archiving directory for storage, and a new log file is created in the logging directory. You cannot change the environment variable which is set by default that is, %ORAWEB_ADMIN%/website40/log. You must give a complete directory path name if you are not using the environment variable, for example, /private/OAS/logs/.

Configuring XLF Logging

The XLF logging feature can generate statistics of HTTP requests, such as the IP address of clients, methods of the HTTP requests, or the URL. It also logs HTTP requests and response headers such as *user agent* and *accept*. These logs are part of the Extended LogFile Format (XLF) which is a superset of the Common LogFile Format (CLF).

CLF/XLF Logging

The Logger can store XLF Logging information in an Oracle database or file system. Later, the Log Analyzer can use the XLF Logging data to present relevant statistics about your site.

The CLF/XLF logging consists of the following:

- [Columns in the Logxlf table](#)
- [Columns in the Logxlfheader table](#)
- [Xlfseq Sequence](#)

Table 10–4 *Columns in the Logxlf table*

Column	Data Type	Description
entry_date	date	The time and date the log entry is recorded.
usec	number	The time (in microseconds) of the log entry.
wrb	varchar2(100)	Indicates a Web Request Broker (WRB) specific value. Currently, it is planned to hold the name of the machine where the WRB resides. This field is used in distributed environments to distinguish between the different WRBs.

Column	Data Type	Description
component	varchar2(2000)	Indicates the component name such as adapter (ADP), virtual path manager (VPM), or resource manager (RM).
id	number	The process id of the process to be logged.
xlfsseq	number	Allows matching of an entry in the logxlf table with the entries in the logxlfheader table.
time_taken	number	The time taken for the transaction to complete
bytes	number	Content-length of the transferred document.
cached	number	Record whether a cached hit occurred.
c_ip	varchar2(100)	Client's IP address and port.
c_dns	varchar2(100)	Client's DNS name.
s_ip	varchar2(100)	Server's IP address and port.
s_dns	varchar2(100)	Server's DNS name.
sc_status	number	Server-to-client status code.
c_auth_id	varchar2(100)	Username if the request contained an attempt to authenticate.
sc_comment	varchar2(100)	Comment returned with status code.
cs_method	varchar2(100)	Server-to-client method (For example, GET, POST).
cs_uri_stem	varchar2(2000)	Client-to-server stem portion of URI, omitting the query.
cs_uri_query	varchar2(2000)	Client-to-server query portion of URI, omitting the stem.

Table 10–5 Columns in the Logxlheader table

Column	Data Type	Description
xlseq	number	Allows matching of an entry in the logxl table with the entries in the logxlheader table.
prefix	varchar2(2)	Indicates if it is a request header “cs” or response header “sc”.
hname	varchar2(2000)	Name of the HTTP header.
hvalue	varchar2(2000)	Value of the HTTP header.

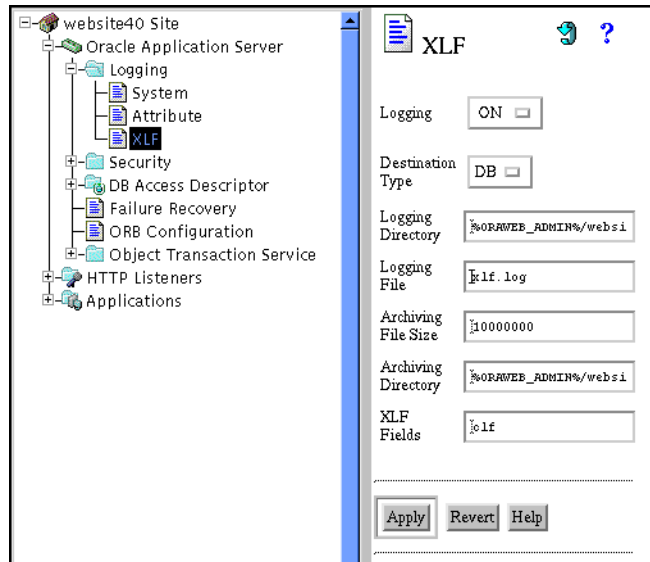
Xlseq Sequence

The sequence is used to match up an entry in the logxl table with the entries in the logxlheader table.

```
create sequence xlseq
  start with 1000
  nomaxvalue
  order;
```

To access the XLF logging form, expand the Logging folder and select XLF:

Figure 10–6 XLF form



Logging

Specifies whether or not to use the Extended LogFile Format (XLF) for logging information to the file. Most Web servers offer the option to store log files in either the common log format or a proprietary format. XLF is an improved format for Web server logging since it is extensible, permitting a wider range of data to be captured. Possible values are:

- ON - Use XLF logging. This is the default.
- OFF - Do not use XLF logging. If you choose not to use XLF Logging, you will only get those fields logged that are part of the Common Logging Format, (CLF). Thus the **xlf.log** (File or the database table) will not be created with these XLF entries.

Destination Type

Indicates the type of storage in which to log the information. Possible values are:

- FS - Use a file system as the storage type. This is the default.
- DB - Use a database as the storage type.

If you want to log files to a database, you must enter a DAD in the general Logging form. See [“Logger DAD Name” on page 10-3](#).

When you choose database (DB) as your storage of the log files, you do not have to enter any information that would apply for the filesystem (FS), such as: logging directory or logging file. The log information is stored in the LOG-SYMSG table in the database.

Logging Directory

Specifies the name of the directory where the log file is located. You cannot change the environment variable, which is set by default that is, %ORAWEB_ADMIN%/website40/log. You must give a complete directory path name if you are not using the environment variable. For example, /private/OAS/logs/.

Logging File

Specifies the name of the file for logging. The default is **xlf.log**.

Archiving File Size

Specifies the maximum log file size in bytes. Log data is added to a file until it reaches the specified size; a new file is then opened, and the old file is archived in the archive directory. For example, if the maximum log file size is specified as 1000, when the log file reaches 1000 bytes, it is archived and a new log file is created.

The default is 10,000,000 bytes (approximately 10 MB).

Note for Unix users: If you have configured multiple Loggers to write to the same file, archiving occurs when one of the Loggers has written 1000 bytes to the file. In such a scenario, it is possible that the size of the file is greater than 1000 bytes when it is archived.

Note for Windows users: Multiple Loggers cannot write to the same log file. If you specify the same log file name for more than one Logger, the first Logger writes data to that file. Other Loggers write their data to *logfile_name.PID*, where PID is the process ID of that Logger.

The archived file name is *logfile_name.n*, where *n* is a number from 0 to 999, inclusive. For example, if your log file is called **xlf.log**, archived files are named: **xlf.log.0**, **xlf.log.1**, ..., **xlf.log.999**. After 1000 log files are archived, **xlf.log** will no longer be archived. All messages will continue to be written to **xlf.log**, regardless of the file size. The Logger does not overwrite an existing archived file (unless its size is 0 bytes).

Archiving Directory

Identifies the directory in which to store archived log files. When the log file reaches the archiving file size, it is copied to the archiving directory for storage, and a new log file is created in the logging directory. You cannot change the environment variable which is set by default that is, %ORAWEB_ADMIN%/website40/log. You must give a complete directory path name if you are not using the environment variable, for example, **/private/OAS/logs/**.

XLF Fields

Specifies the specific fields to log. These fields are part of the Extended Log File Format (XLF) which is a superset of the Common LogFile Format (CLF). To specify the values, enter the fields separated by spaces.

The default is `clf`.

[Table 10–6](#) describes the fields:

Table 10–6 Description of fields

Field	Description
clf	Fields written to a CLF file include: c-dns cauth-id [clf-date] “ request line ” sc-status bytes
cs (User-Agent)	Information about the user agent originating the request
cs (Referer)	Allows the client to specify the address (URI) of the resource from which the Request-URI was obtained
clf-date	Date and time of the request in the same format as used in the CLF: [dd/mm/yyyy:hour:minutes:seconds GMT_offset]
c-auth-id	Username if the request contained an attempt to authenticate
bytes	Content-length of the transferred document

Field	Description
date	Date at which the transaction completed
time	Time at which the transaction completed
time-taken	Amount of time taken (in seconds) for transaction to complete
c-ip	Client's IP address and port
s-ip	Server's IP address and port
c-dns	Client's DNS name
s-dns	Server's DNS name
sc-status	<p>Server-to-client status code</p> <p>The first digit of the status code defines the class of response. The last two digits do not have any categorization role. There are 5 values for the first digit:</p> <p>1xx: Informational - Request received, continuing process</p> <p>2xx: Success - The action was successfully received, understood, and accepted</p> <p>3xx: Redirection - Further action must be taken in order to complete the request</p> <p>4xx: Client Error - The request contains bad syntax or cannot be fulfilled</p> <p>5xx: Server Error - The server failed to fulfill an apparently valid request</p> <p>The following is a partial list of the status codes:</p> <p>“101” Switching Protocols</p> <p>“200” OK</p> <p>“300” Multiple Choices</p> <p>“400” Bad Request</p> <p>“500” Internal Server Error</p> <p>For more information about status codes, see http://www.w3.org/Protocols</p>

Field	Description
sc-comment	Server-to-client comment returned with the status code
cs-method	Server-to-client method (GET, POST, or others)
cs-protocol	HTTP protocol version
cs-uri	Client-to-server URI
cs-uri-stem	Client-to-server stem portion of URI, omitting the query
cs-uri-query	Client-to-server query portion of URI, omitting the stem
prefix (header)	Header is an HTTP header field and prefix is one of the following: <ul style="list-style-type: none"> ■ c: client ■ s: server ■ r: remote ■ cs: client-to-server ■ sc: server-to-client

For information about viewing the XLF logging entries, see [“XLF Log Entries” on page 10-31](#).

Installing the Utilities

Oracle Application Server provides the following logging and database utilities:

- PL/SQL Toolkit
- Log Analyzer
- Database Browser

These utilities depend on one another to function, and must be installed in the proper order.

All of the HTML forms for installing the database and logging utilities are accessible from the OAS Utilities site. To access this site (once you have successfully installed Oracle Application Server):

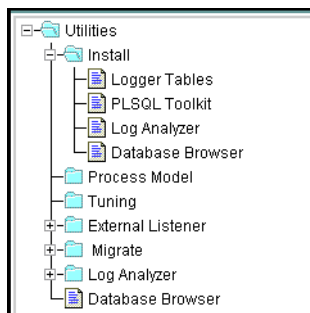
1. Connect to the Welcome page using your browser. (See [“Accessing the Welcome Page” on page 1-5](#) for more information.)

2. Select OAS Utilities. The OAS Utilities navigational tree appears.

Note: The password you entered for the Node Manager listener during installation is also the password for the Administration Utility listener.

3. Expand the Utilities folder.
4. Expand the Install folder.

Figure 10–7 OAS Utilities



Each task below discusses the forms listed in the Install folder.

Before You Install

Before you install the utilities:

- Your database must be installed and running, including the listener.
- You will need to select a Database Access Descriptor (DAD) for several of these tasks. You can use an existing DAD, or create a new one during installation. For more information, see [“Creating a DAD” on page 6-2](#).

Task 1: Install the Logger Tables

The Logger tables are required if you want to write log information to a database. Storing this information in a database enables you to use the Log Analyzer to run reports on the log information.

Note: If you choose not to install the Logger tables, you can still generate log files, but you will not be able to generate reports on the data using the Log Analyzer.

1. Click “Logger Tables” in the Install folder to display the Install Logger form.

Figure 10–8 *Install Logger Tables Form*



2. Select a DAD from the drop-down listbox and click Apply.
(If there are no DADs in the list, you must create one by clicking the Create New DAD button.)
The DBA Account Info screen appears.
3. Enter the SYS password and click Apply.

Note: If you are using an Oracle8 database, you may get an error that the **catdbsyn.sql** script cannot be opened. You can ignore this error and continue. The script is used with Oracle7.

A window pops up displaying messages about the installation of the LOGGER packages.

4. Check the messages for any errors and click OK.

Task 2: Install the PL/SQL Toolkit

The PL/SQL Toolkit is required if you want to:

- use the PL/SQL samples
- use the Log Analyzer

- use the Database Browser
- use a database with Oracle Application Server

To install the PL/SQL Toolkit:

1. Click the PL/SQL Toolkit icon in the Install folder to display the Install PL/SQL Toolkit form.

Figure 10–9 Install PL/SQL Toolkit form

The screenshot shows the 'Oracle Application Server Utilities' window. On the left is a tree view under 'Utilities' with 'Install' expanded, showing 'Logger Tables', 'PL/SQL Toolkit' (highlighted), 'Log Analyzer', and 'Database Browser'. The main area is titled 'Install' and contains the following fields: 'ORACLE_SID' (empty), 'Connect String' (empty), 'DBA Username' (set to 'sys'), 'Password' (empty), and 'Confirm Password' (empty). At the bottom are 'Apply', 'Revert', and 'Help' buttons.

2. Fill in the form's fields:

(You must specify an ORACLE_SID or a connect string.)

- **ORACLE_SID** - If your database is local (installed in the same ORACLE_HOME as Oracle Application Server), enter the database SID.
For example, ORCL.
- **Connect String** - If your database is remote (not installed in the same ORACLE_HOME as Oracle Application Server), enter the connect string as defined in your **tnsnames.ora** file.

Note: Entering an incorrect connect string can cause many problems for the OAS Utilities. For a complete discussion about creating and testing a connect string, see the section [“Create a connect string.”](#) on page 6-2.

- **DBA Username** - This field is not editable. The **sys** user has administration privileges on the database.
- **Password** - Enter the password for the **sys** user.

Note: When you install the PL/SQL Toolkit, you do not select a DAD. The PL/SQL packages are automatically installed under the default user **oas_public**.

A window pops up displaying messages about the installation of the PL/SQL and Content Services packages.

3. Check the messages for any errors and click OK.

Task 3: Install the Log Analyzer

The Log Analyzer allows you to analyze the raw data stored by the Logger. For example, the Log Analyzer can display statistics such as total bytes transferred or the most frequently accessed URL.

Note: The Log Analyzer depends on the PL/SQL Toolkit and the Logger tables to function. You must install all three components to use the Log Analyzer.

To install the Log Analyzer:

1. Click the Log Analyzer icon in the Install folder to display the Install form.
2. Select a DAD from the drop-down listbox and click Apply.

The DBA Account Info screen appears.

3. Enter the SYS password and click Apply.

A window pops up displaying messages about the installation of the Log Analyzer packages.

4. Check the messages for any errors and click OK.

For directions on how to use the Log Analyzer, see [“Using the Log Analyzer” on page 10-28](#).

Task 4: Install the Database Browser

The Database Browser allows you to view:

- Tables and views
- Stored PL/SQL procedures, functions, and packages
- Database users
- Database sessions
- Rollback segments
- Database files
- Tablespace space usage
- Tablespace free space summary

Note: The Database Browser depends on the PL/SQL Toolkit and the Logger tables to function. You must install all three components to use the Database Browser.

To install the Database Browser:

1. Select Database Browser.
2. Select a DAD and click Apply.

The DBA Account Info screen appears.

3. Enter the SYS user password and click Apply.

A window pops up displaying messages about the installation of the Database Browser packages.

4. Check the messages for any errors and click OK.
5. Use the OAS Manager to associate this DAD with the Database Browser cartridge.
 - a. Return to the Welcome page by clicking the “Go to OAS Home” button.
 - b. From the Welcome page, select OAS Manager.
 - c. Expand the Site folder.
 - d. Expand the Applications folder.

- e. Expand the DB Utilities folder.
- f. Expand the Cartridges folder.
- g. Expand the DB Browser folder.
- h. Expand the Configuration folder and select PL/SQL Parameters.

The PLSQL Agent Configuration form appears:

Figure 10–10 PL/SQL Agent Configuration form

PLSQL Agent Configuration

DAD: ltdb

Protection: Disabled

HTML Error Page: error.html

Error Level: Advanced

Compress Uploaded Files: No

Session Abort URI: I

Buttons: Apply, Revert, Create New DAD, Help

- i. From the DAD drop-down listbox, select the DAD you used to install the Database Browser and click Apply.

Note: You must configure a DAD in the PL/SQL Parameters form for each cartridge under the DB Utilities application before you can run any of the cartridges in that application.

For directions on how to use the Database Browser, see [“Using the Database Browser” on page 10-48](#).

Task 5: Grant Privileges to the DAD User

The user specified in the DAD used to install the utilities must have the proper roles. Once you have installed all of the utilities, grant the following privileges as the SYS user:

```
SVRMGR> grant ows_standard_role to <user>
SVRMGR> grant all on websys.owa_content to <user>
```

Using the Log Analyzer

Overview

Oracle Application Server gives you the option of storing your log files, such as the **wrb.log** or **xlif.log** in a table in a database. You can then use the Log Analyzer to view this information through your browser. The Log Analyzer also allows you to run customizable reports on the data. If you prefer to store the log files in a filesystem, you can easily upload them into the database for use with the Log Analyzer.

The information that the Log Analyzer provides is important in determining how the contents of a Web site are being accessed, which links are repeatedly accessed, or who are the most common visitors. This information can help you organize the Web site information in a more efficient manner.

Note: You cannot use the Log Analyzer to look at data that was written to the database using `WRB_LOGwriteAttribute()`.

The Log Analyzer stores each log file in the predefined database table — `logreport`:

Table 10-7 Columns in the Logreport table

Column	Data Type	Description
rtype	varchar2(100)	The analyzer report name.
rdescription	varchar2(2000)	The analyzer report description.
qdescr	long	Indicates the SQL query of the analyzer report.
rform	varchar2(2000)	Indicates the analyzer report form values.
dimension	number	Indicates the number of output entries to display.

Uploading Log Files into the Database

Each of your log files will take the form of either an XLF type, a system message type (SYSSMSG), or a client defined attribute type (ATTRIB). Whenever you pick a file to upload to the database, you must select one of these formats so that Oracle Application Server knows what table to use and also how to handle the data for import into that table. Each of the files that you want to upload will have to be uploaded separately and can be only of the type XLF, SYSSMSG, or ATTRIB.

For logging configuration and setup, see [“Configuring Logging Features” on page 10-1](#).

To view the Database Upload screen:

1. From the Welcome page, select OAS Utilities.
2. Expand the Utilities folder.
3. Expand the Log Analyzer folder.
4. Select Database Upload.

The following sections describe the Database Upload form’s fields.

Figure 10–11 Database Upload form

The screenshot shows the Oracle Application Server Utilities interface. On the left is a tree view under 'Utilities' containing 'Install', 'Process Model', 'Tuning', 'External Listener', 'Migrate', 'Log Analyzer', 'Database Upload' (highlighted), 'Log Entries', 'Log Report', and 'Database Browser'. The main area is titled 'Database Upload' and contains the following fields: 'DAD name:' with a pull-down menu, 'Path:' with a text box containing '/private1/408/ows/admin/w', 'File:' with a text box containing 'l', and 'File type:' with a pull-down menu showing 'XLF'. At the bottom are four buttons: 'Truncate', 'Upload', 'Revert', and 'Help'.

DAD name

A pull-down menu to select the DAD to use. Make sure to use the same DAD name (DBA) you selected in the Logging form. See the section [“Logger DAD Name” on page 10-3](#).

Path

Enter the complete path to the file name containing the XLF or system messages. For example:

UNIX: **/ows/admin/website40/log**

NT: **D:\ows\admin\website40\log**

File

Enter the file name containing the XLF, attribute, or system messages. For example, **xlf.log** for XLF messages.

File Type

A pull-down menu to specify the file type. Possible values are:

- XLF - The file type to upload Extended LogFile Format (XLF).
- SYMSG - The file type to upload system messages.

- ATTRIB - The file type to upload client-defined attributes.

After filling in the necessary information, click the 'Upload' button to upload the log file into the database. The upload can take a few minutes depending on the size of the file.

The 'Truncate' button allows you to avoid duplicate log entries in the table. This option can be used if a log file has already been uploaded, but some changes have been made.

Viewing Log Entries

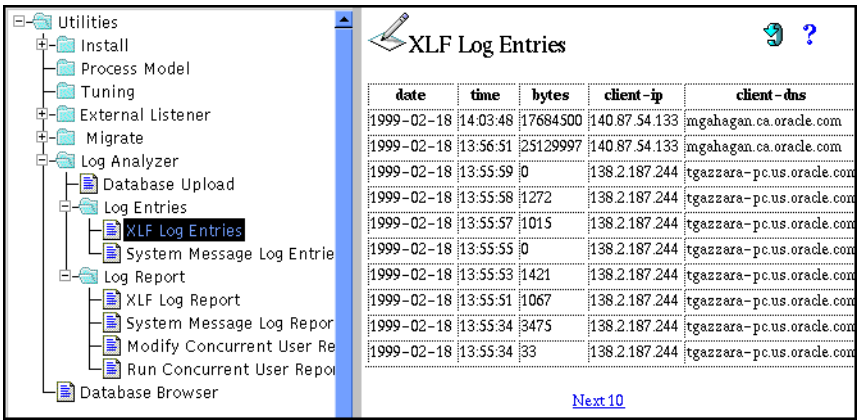
XLF Log Entries

To view the contents of the uploaded XLF file:

1. Expand the Log Analyzer folder.
2. Expand the Log Entries folder.
3. Select XLF Log Entries.

The XLF Log Entries form displays a sample XLF log file. The table displays the last ten log entries. Null entries contain the '-' character.

Figure 10–12 XLF Log Entries form



date	time	bytes	client-ip	client-dns
1999-02-18	14:03:48	17684500	140.87.54.133	mgahagan.ca.oracle.com
1999-02-18	13:56:51	25129997	140.87.54.133	mgahagan.ca.oracle.com
1999-02-18	13:55:59	0	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:58	1272	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:57	1015	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:55	0	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:53	1421	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:51	1067	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:34	3475	138.2.187.244	tgazzara-pc.us.oracle.com
1999-02-18	13:55:34	33	138.2.187.244	tgazzara-pc.us.oracle.com

Table 10–8 describes the XLF Log entries.

Table 10–8 Columns in the XLF Log Entries form

Column	Description
date	Date of the request in the format [yyyy-mm-dd]
time	Time of the request in the format [hour:minutes:seconds]
bytes	Content-length of the transferred document
client-ip	Client's IP address and port
client-dns	Client's DNS name
	If the entries in these columns are NULL '-', you need to set the Web Listener to resolve IP addresses into hostnames.
server-ip	Server's IP address and port
server-dns	Server's DNS name
status	Server-to-client status code The first digit of the status code defines the class of response. The last two digits do not have any categorization role. There are five values for the first digit: <ul style="list-style-type: none">■ 1xx: Informational - Request received, continuing process■ 2xx: Success - The action was successfully received, understood, and accepted■ 3xx: Redirection - Further action must be taken in order to complete the request■ 4xx: Client Error - The request contains bad syntax or cannot be fulfilled■ 5xx: Server Error - The server failed to fulfill an apparently valid request For more information about status codes, see http://www.w3.org/Protocols
method	Server-to-client method (for example, GET, POST)
uri-stem	Client-to-server stem portion of URI, omitting the query
uri-query	Client-to-server query portion of URI, omitting the stem
auth-id	Username if the request contained an attempt to authenticate

To see more entries, click the Next 10 link.

System Messages Log Entries

To view the contents of the uploaded system message log file:

1. Expand the Log Analyzer folder.
2. Expand the Log Entries folder.
3. Select System Message Log Entries.

The Log Analyzer begins to load and format the file.

The System Messages Log Entries form displays the stored log entries. The table displays the last ten log entries. Null entries contain the '-' character.

Figure 10–13 System Messages Log Entries form

System Messages Log Entries							
The last 10 log entries are:							
date	time	wrb	component	id	severity	mask	message
1999-04-21	08:05:06	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	5238	10	4198399	Logging on as user: joe
1999-04-21	08:05:05	dstahler-sun	Dispatcher	5182	13	12287	OWS-04526: For tracing only: no exception after execute_nw.
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Request Context created
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Adding the cartridge OR to the thread context
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Running Post Recv filters
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Response Context created
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Authorizing the request
1999-04-21	08:05:05	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	5238	8	4198399	Authorizing request by agent: DB_Uilities/Log_Analyzer
1999-04-21	08:05:05	dstahler-sun	Cartridge Server	5238	13	4198399	OWS-10860: CS Debug: Invoking the Exec Callback
1999-04-21	08:05:05	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	5238	8	4198399	Invoking anlz.listsyslog ...
Next 10							

The following table describes the entries made in the system messages log entries form:

Table 10–9 Columns in the System Messages Log Entries form

Column	Description
date	Date of the request in the format [yyyy-mm-dd]
time	Time of the request in the format [hour:minutes:seconds]
wrb	Indicates a Web Request Broker (WRB) specific value. It holds the name of the machine where the WRB resides. This field is used in a distributed environment to distinguish between the different WRBs.
component	The component name. For example, PL/SQL, virtual path manager, resource manager, log module, dispatcher, or authentication.
id	Process id of the process that is to be logged.
severity	Severity of the system message.
mask	The mask that gives the capability to log according to a component or service within the WRB cartridge component.
message	Generic system message.

To see more entries, click the Next 10 link.

Running Log Reports

The Log Analyzer allows you to run pre-determined or user-defined reports on the log files. To display the Log Report forms:

1. From the Welcome page, select OAS Utilities.
2. Expand the Utilities folder.
3. Expand the Log Analyzer folder.
4. Expand the Log Report folder.

The four Log Report options are displayed:

- [XLF Log Report](#)
- [System Message Report](#)

- [Modify Concurrent User Report](#)
- [Run Concurrent User Report](#)

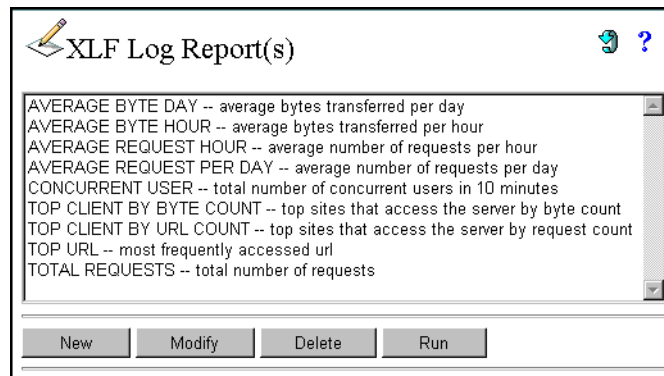
Select one of the options to display a form.

XLF Log Report

There are several different reports which you can run on the XLF log file, whether it was uploaded from a file or logged directly into the database.

When you select XLF Log Report, the form is displayed:

Figure 10-14 XLF Log Report(s) form



XLF Log Report(s)

- AVERAGE BYTE DAY -- average bytes transferred per day
- AVERAGE BYTE HOUR -- average bytes transferred per hour
- AVERAGE REQUEST HOUR -- average number of requests per hour
- AVERAGE REQUEST PER DAY -- average number of requests per day
- CONCURRENT USER -- total number of concurrent users in 10 minutes
- TOP CLIENT BY BYTE COUNT -- top sites that access the server by byte count
- TOP CLIENT BY URL COUNT -- top sites that access the server by request count
- TOP URL -- most frequently accessed url
- TOTAL REQUESTS -- total number of requests

New Modify Delete Run

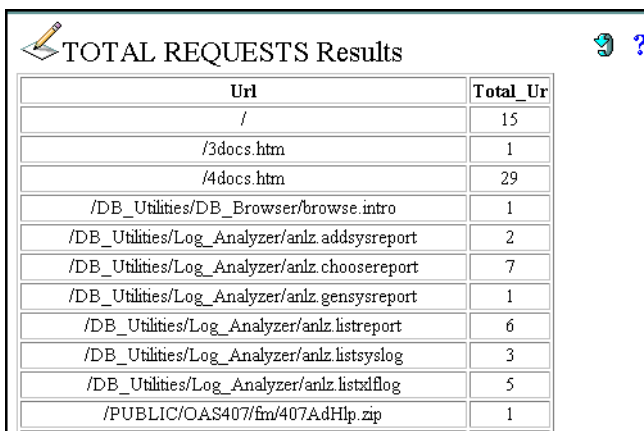
This form can be used for:

- [Running a Report](#)
- [Deleting a Report](#)
- [Creating or Modifying a Report](#)

Running a Report To run a selected report, select it and click the 'Run' button.

For example, if you select TOTAL REQUESTS and click Run, you will see a report similar to the following:

Figure 10–15 *Total Requests report*



Url	Total_Ur
/	15
/3docs.htm	1
/4docs.htm	29
/DB_Uilities/DB_Browser/browse.intro	1
/DB_Uilities/Log_Analyzer/anlz.addsysreport	2
/DB_Uilities/Log_Analyzer/anlz.choosereport	7
/DB_Uilities/Log_Analyzer/anlz.gensysreport	1
/DB_Uilities/Log_Analyzer/anlz.listreport	6
/DB_Uilities/Log_Analyzer/anlz.listsyslog	3
/DB_Uilities/Log_Analyzer/anlz.listdflog	5
/PUBLIC/OAS407/fm/407AdHlp.zip	1

Deleting a Report To delete a report from the report(s) list, select the report you want to delete and click the 'Delete' button.




Note: You cannot confirm your deletion; therefore, be sure you are selecting the correct report before you delete it.

Creating or Modifying a Report To modify or create a report for generating user-defined statistics:

1. Click the New or Modify button.

The Add/Modify XLF Report form appears:

Figure 10–16 Add/Modify XLF Log Report form

 Add/Modify XLF Report  

Report Name

Report Description

Field Name	Display	Operation	Comparator	Comparison Value
Client IP Address	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Client Host	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Url	<input type="text" value="Yes"/>	<input type="text" value="Total"/>	=	<input type="text"/>
Bytes	<input type="text" value="No"/>	<input type="text" value="Null"/>	= <input type="text"/>	<input type="text"/>
Status	<input type="text" value="No"/>	<input type="text" value="Null"/>	= <input type="text"/>	<input type="text"/>
Method	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Domain	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Auth Id	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Server IP Address	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Server Host	<input type="text" value="No"/>	<input type="text" value="Null"/>	=	<input type="text"/>
Date	<input type="text" value="No"/>	<input type="text" value="Null"/>	From <input type="text"/>	<input type="text"/>
			To <input type="text"/>	

The following sections describe the parameters that you can configure using the Add or Modify XLF Report form.

Fields in the top of the form:

- **Report Name** - You can create or alter the report name here.
- **Report Description** - This field allows you to create or alter the description of the report.

In the middle section of the form, there are columns consisting of field types, and rows consisting of the parameters. The field types are listed here:

- **Field Name** - Specifies the report field names. These fields are used to generate the columns of your report. For example, Client IP Address.
- **Display** - Allows you to select whether or not you want a particular field to be a part of the report. Possible values are 'Yes' and 'No'. The default value for each field is 'No'. For example, if you select 'Yes' for the Date and URL fields, then the report page will contain two columns (Date and URL).
- **Operation** - Applies an operation on the report field. The operation applied depends on the particular field. Any time there is an operation applied to a field, a column representing the operation on the field appears in the report page. The operation may also be 'Null'. In this case, an operation will not be applied.
- **Comparator** - Specifies how the fields in the log entries are compared against the value field. Numerical fields can contain a '<', '>', and '=' comparator. String fields have only '=' comparator.

Comparison Value - Specifies the value used to compare against the fields. For example, to query all the log entries with domains that contain .gov, you can enter "gov" in the Comparison Value column in the Domain row.

You can also input wildcards in the comparator value field. For example, "*.com" in the client host field will match up any hostname that ends in ".com". The "*" will match zero or more characters. Except for time, date, bytes, domain, and country fields, wildcards are allowed.

The following table describes the field names and operations you can use for each field:

Table 10–10 Columns in the Add/Modify xlf Log report form

Field Name	Description	Operation
Client IP Address	Client's IP address	■ Null
		■ Total
		■ Average
Client Host	Hostname of the client or the browser sending the HTTP request. It is available only if the listener has enabled DNS resolution.	■ Null
		■ Total
		■ Average

Field Name	Description	Operation
Url	Represents the uri_stem field only. It does not include the uri_query, which is everything after the question mark in the URL.	■ Null
		■ Total
		■ Average
Bytes	Content-length of the transferred document in bytes. The bytes field has three comparator choices: '=', '>', or '<'. Therefore, you can generate statistics such as all requests that transfer > 500 bytes. Note that this field query applies to the Display column (either 'yes' or 'no').	■ Null
		■ Total
		■ Average
		■ Minimum
		■ Maximum
Status	HTTP status code for the request. The status field also has three comparator choices: '=', '>', or '<'. Note that this field query applies to the Display column (either 'yes' or 'no').	■ Null
		■ Total
		■ Average
Method	Server-to-client method. For example, GET, POST.	■ Null
		■ Total
		■ Average
Domain	Three letter suffix representing the domain name of the client, for example, 'com', 'edu', 'mil', 'gov', 'net', and 'org'.	■ Null
		■ Total
		■ Average
Auth Id	The user ID used for authentication during the user's request.	■ Null
		■ Total
		■ Average
Server IP Address	Server's IP address on which the sever is listening for HTTP requests. For example '144.25.99.138'.	■ Null
		■ Total
		■ Average

Field Name	Description	Operation
Server Host	Name of the host.	<ul style="list-style-type: none">■ Null■ Total■ Average
Date	Represents the XLF log entry date. The date field has two comparators: From and To. For example, you can generate a list of the total number of URLs from 1997-10-01 to 1998-09-15. The format for the date is 'yyyy-mm-dd'.	<ul style="list-style-type: none">■ Null■ By Year■ By Month■ By Day
Time	Time at which the XLF log is entered in the database. The time field has two comparators: From and To. The time format is 'hh:mi:ss'. The hour is in the 24-hour format. For example, 18:10:00.	<ul style="list-style-type: none">■ Null■ By Hour - Valid range is 0 to 23.■ By 10 minute

You can also input wildcards in the comparator value field. For example, '*.com' in the client host field will match up any hostname that ends in '.com'. The '*' will match zero or more characters. Wildcards are allowed in the Comparator Value field except for time, date, bytes, domain, and country fields.

The fields in the bottom of the form:

- **Dimension** - Allows you to display a fixed number of rows. For example, 15. If the dimension is set to 0, the report generated does not display any rows.
- **Sort By** - Sorts a particular operation or report field. For the same field, if both operations are selected and display = 'yes', the report will be sorted according to the operation. Note that it is illegal to sort a field where display = 'no' and operation = 'Null'. There is no need to sort in this case.
- **Sort Order** - Specifies the order in which to sort the results, such as: ascending or descending. After selecting the 'total' for a particular field, the 'sort by' is selected for that same field, and the 'sort order' can be selected.
- **Distinct** - Specifies whether or not to display distinct entries.
- **Save Report** - Click the Save Report button to save the current report.
- **Reset Form** - Click the Reset Form button to revert all fields to their defaults.

The following are some examples:


Table 10–11 Examples


Total number of requests	<ul style="list-style-type: none"> ■ URL Display = 'Yes' ■ URL Operation= 'Total' ■ Leave the Comparison Value field blank
Average requests per day	<ul style="list-style-type: none"> ■ URL Operation= 'Average' ■ Date Operation= 'By Day'
Most common Urls	<ul style="list-style-type: none"> ■ Url Display = 'Yes' ■ Url Operation = 'Total' ■ Sort By = 'Url' ■ Sort Order= 'Descending'
Top sites that access the server by request count	<ul style="list-style-type: none"> ■ Client Host Display = 'Yes' ■ Url Operation = 'Total' ■ Sort By = 'Url' ■ Sort Order = 'Descending'
Top sites that access the server by bytes transferred	<ul style="list-style-type: none"> ■ Client Host Display = 'Yes' ■ Bytes Operation = 'Total' ■ Sort By = 'bytes' ■ Sort Order = 'Descending'
Top two dates between 1997-10-01 and 1998-09-15 by total number of bytes transferred	<ul style="list-style-type: none"> ■ Date Display = 'Yes' ■ Date Comparator From = '1997-10-01' ■ Date Comparator To = '1998-09-15' ■ Bytes Operation = 'Total' ■ Dimension = '2' ■ Sort By = 'Bytes' ■ Sort Order = 'Descending'

System Message Report

The System Messages Report(s) form (Figure 10-17) allows you to filter the data that the report returns.

Figure 10-17 System Messages Report(s) form

System Messages Report(s)

?

Field Name	Comparator	Value
Date	From	<input type="text"/>
	To	<input type="text"/>
Time	From	<input type="text"/>
	To	<input type="text"/>
Component Name	=	<input type="text"/>
Process Id	=	<input type="text"/>
Severity Level	= <input type="button" value="v"/>	<input type="text"/>

Dimension

You can create your own set of queries to trace a specific component that is providing information to the Logger, or you can select Submit to query all the data.

This section describes the form's fields:

- **Field Name** - Specifies the report field names. These fields are used to generate the columns of your report.
- **Comparator** - Specifies how the fields in the log entries are compared against the value field. Numerical fields contain '<', '>', and '=' comparator. String fields have only '=' comparator.
- **Value** - Specifies the value used to compare against the fields.

Table 10-12 describes the values you can use for each field.

Table 10–12 Columns in the System Messages Report(s) form

Field Name	Description	Comparator/Value
Date	Date of the request	The date field has two comparators: 'From' and 'To'. For example, you can generate a report from 1997-10-01 to 1998-09-15. The format for the date is 'yyy-mm-dd'.
Time	Time of the request	The time field has two comparators: 'From' and 'To'. The format for the time is 'hh:mi:ss'. The hour is in the 24 -hour format.
Component Name	Name of the component	The component name. For example, PL/SQL, virtual path manager, resource manager, log module, dispatcher, or authentication.
Process Id	Process id of the process to be logged	The Process Id field has one comparator '='.
Severity Level	Severity of the system message	The Severity Level field has three comparator choices: '=', '>', or '<'. Therefore, you can generate statistics, such as: all the severity levels > 8. The possible values are 0 through 15.

- **Dimension** - Allows you to display a fixed number of rows. For example, 15. If the dimension is 0, the report generated does not display any rows.
- **Submit Report** - After filling in the necessary information, click the 'Submit Report' button to run the report. If you need to Reset the form, you can use the 'Reset Form' button.

A sample System Messages Report Results form is shown in [Figure 10–18](#).

Figure 10–18 System Messages Reports Results

System Messages Report Results							
EntryDate	EntryTime	Wrb	Component	Id	Severity	Mask	Message
1999-04-19	20:16:06	dstahler-sun	Cartridge Server	3842	13	4198399	OWS-10860: CS Debug Running Post Recv filter
1999-04-19	20:16:06	dstahler-sun	Cartridge Server	3842	13	4198399	OWS-10860: CS Debug Authorizing the request
1999-04-19	20:16:06	dstahler-sun	Cartridge Server	3842	13	4198399	OWS-10860: CS Debug Invoking Auth Callback.
1999-04-19	20:16:06	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	3842	8	4198399	Authorizing request by agent: DB_Uilities/Log_Analy
1999-04-19	20:16:06	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	3842	8	4198399	Invoking anlz gensysrep ...
1999-04-19	20:16:06	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	3842	10	4198399	Logging on as user: dav
1999-04-19	20:16:06	dstahler-sun	DB_Uilities/Log_Analyzer Cartridge	3842	8	4198399	Executing request by age DB_Uilities/Log_Analy
1999-04-19	20:16:06	dstahler-sun	Cartridge Server	3842	13	4198399	OWS-10860: CS Debug Invoking the Exec Callba

Table 10–13 describes the entries.

Table 10–13 Columns in the System Messages Reports Results form

Column	Description
Entry Date	Date of the request in the format [yyyy-mm-dd].
Entry Time	Time of the request in the format [hour:minutes:seconds].
Wrb	Indicates a Web Request Broker (WRB) specific value. It holds the name of the machine where the WRB resides. This field is used in distributed environments to distinguish between the different WRBs.
Component	The component name. For example, PL/SQL, virtual path manager, resource manager, log module, dispatcher, or authentication.
Id	Process id of the process to be logged.
Severity	Severity of the system message.

Column	Description
Mask	The mask that gives the capability to log according to a component or service within the WRB cartridge component.
Message	Generic system message.

Modify Concurrent User Report

A useful indication of the performance and stability of your website is the number of requests it can service at one time. The CONCURRENT USER report allows you to gather this information.

By default, the Concurrent User report is measured within a 10 minute interval and specifies the number of distinct IP addresses that appear in the XLF log entry.

If two users are going through the same firewall (within the specified interval), they will be counted as one, since the client is the proxy.

The Modify CONCURRENT USER Report form (Figure 10–19) allows you to modify the CONCURRENT USER report:

Figure 10–19 Modify CONCURRENT USER Report form

The screenshot displays the 'Modify CONCURRENT USER Report' form. On the left, a tree view under 'Utilities' shows the navigation path: Utilities > Log Analyzer > Log Report > Modify Concurrent User Report. The main form area on the right includes:

- Period**: Fields for 'From' (Date and Time) and 'To' (Date and Time).
- Time Interval**: A dropdown menu currently set to 'By 10 Minute'.
- Sort By**: A dropdown menu currently set to 'Client IP Address'.
- Buttons for 'Save Report' and 'Reset Form' at the bottom.

The Modify CONCURRENT USER report form allows you to select the fields to create your query.

Period You can narrow the scope of your report by entering specific time restrictions in the Date and Time fields.

- **Date** - Represents the log entry date. The date field has two comparators: From and To. For example, you can generate a list of the total number of concurrent users from 1997-10-01 to 1998-09-15. The format for the date is “YYYY-MM-DD”.

By leaving this field blank the report includes all dates.

- **Time** - The time field has two comparators: From and To. The time format is “HH:MM:SS”. The hour is in the 24-hour format. For example, you can generate a report of the total number of concurrent users from 08:00:00 to 23:59:59.

By leaving this field blank the report is not restricted by time.

Time Interval Here you can select at what time interval the report will group the results. The default is every 10 minutes. Possible values are:

- By 10 Minute
- By Hour
- By Day
- By Month
- By Year
- Null

Sort By This field allows you to determine how to sort the results of the query. Possible values are:

- Time Interval — Sort by the time interval selected in the above field.
- Client IP Address — Sort by the client’s IP address.
- Other — Sort by other fields such as Client Host or URL.

Note: If you need to sort by a particular field, you need to use the XLF Log Report form, select the CONCURRENT USER report, and modify it.

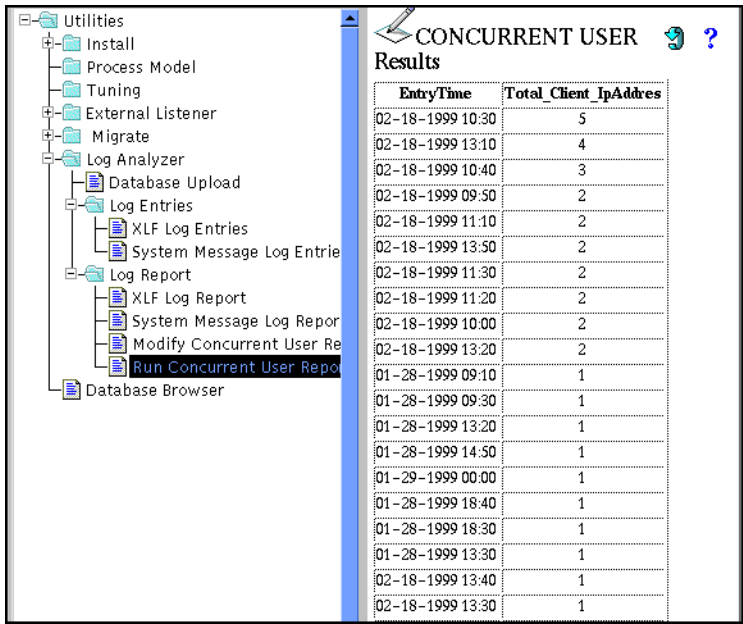
After entering the necessary information, click the Save Report button to save the report.

If you need to reset the form, you can use the Reset Form button.

Run Concurrent User Report

Figure 10-20 shows a sample CONCURRENT USERS report:

Figure 10-20 CONCURRENT USER Results form



Your report may contain only some of these entries depending on the fields you selected to display. This section describes the entries:

Client IP Address Client's IP address

Client Host Hostname of the client or the browser sending the HTTP request. It is available only if the listener has enabled DNS resolution.

Uri Represents the uri_stem field only. It does not include the uri_query which is everything after the question mark in the URL.

Bytes Content-length of the transferred document in bytes

Status HTTP status code for the request

Method Server-to-client method (For example, GET, POST)

Domain Three letter suffix representing the domain name of the client, for example, 'com', 'edu', 'mil', 'gov', 'net', and 'org' or the country name of the client 'uk', 'au'.

Auth Id User ID used for authentication during the user's request

Server IP Address Server's IP address on which the server is listening for HTTP requests, for example, '144.25.99.138'.

Server Host Name of the host

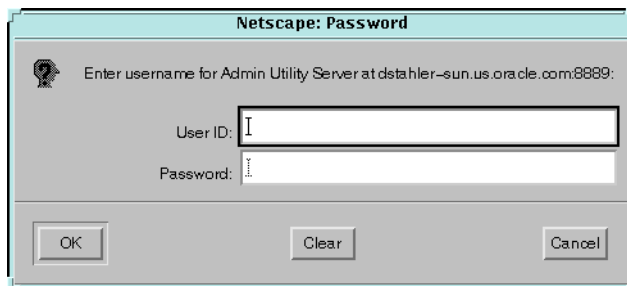
Entry Date Represents the XLF log entry date

Entry Time Time at which the XLF log is entered in the database

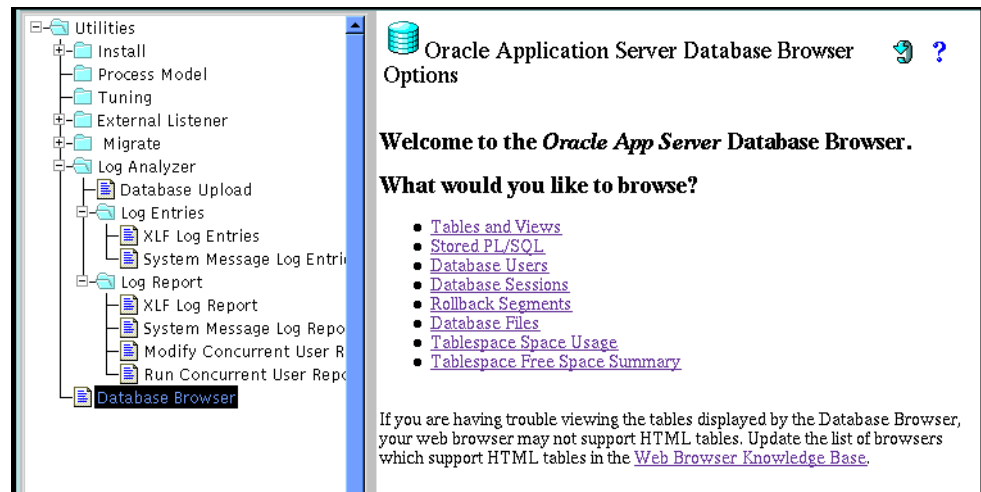
Using the Database Browser

1. From the Welcome page, select OAS Utilities.
2. Expand the Utilities folder.
3. Select Database Browser at the bottom of the list.
4. Enter the username and password for the Admin Utility Server. (By default, these are the same as for the Node Manager.)

Figure 10–21 Admin Utility Server password screen



The Database Browser Welcome page appears.

Figure 10–22 DB Browser Welcome page

Functions of the Database Browser

The Database Browser forms allow you to query and view the following information about your database:

- [Tables and Views](#)
- [Stored PL/SQL](#)
- [Database Users](#)
- [Database Sessions](#)
- [Rollback Segments](#)
- [Database Files](#)
- [Tablespace Space Usage](#)
- [Tablespace Free Space Summary](#)

Note: For a description of these database concepts, refer to your database administration guide.

These options are displayed on the main database browser page. Select the desired topic to proceed.

Note: The first time you use the Database Browser, you are asked whether your Web browser supports HTML tables. This information is used to display Database Browser information in an appropriate format for your Web browser.

No - click No if you use a browser that does not support HTML tables.

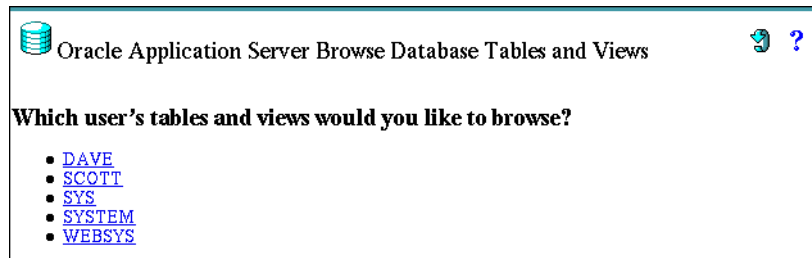
Yes - click Yes if you use a browser with HTML table support.

Not Sure - If you are not sure whether or not your browser supports HTML tables, click Not sure to determine this.

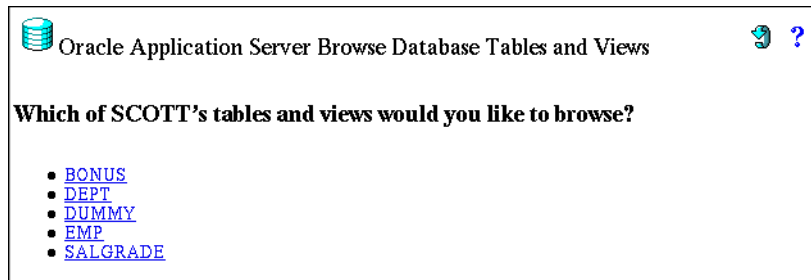
Tables and Views

1. Select Tables and Views to display database tables or views. A list of database users appears, as shown in [Figure 10-23](#).

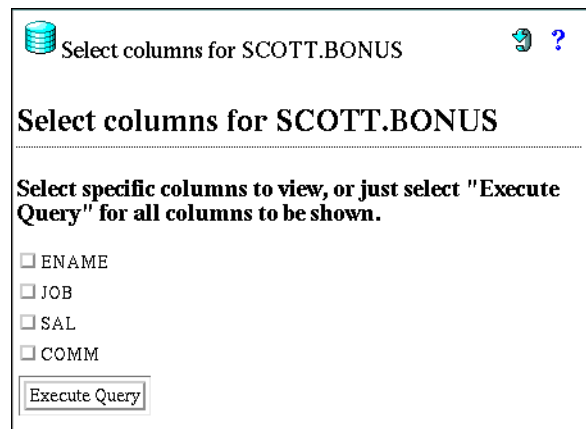
Figure 10-23 *Select User*



2. Click the user whose tables and views you would like to browse. A list of tables and views for that user is displayed as shown in [Figure 10-24](#).

Figure 10–24 Select tables and views

3. Select the table you would like to browse. A list of columns in that table appears.

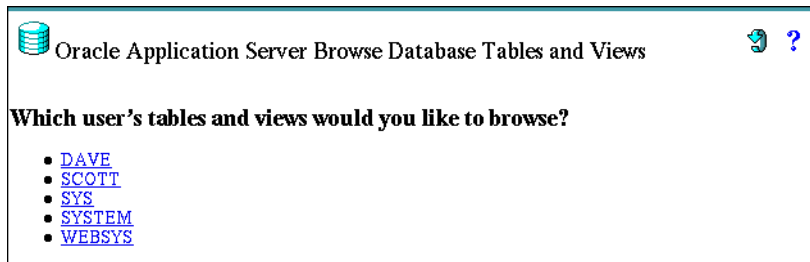
Figure 10–25 Scott's BONUS table

4. Select specific columns to view or, select Execute Query to display all columns.

Stored PL/SQL

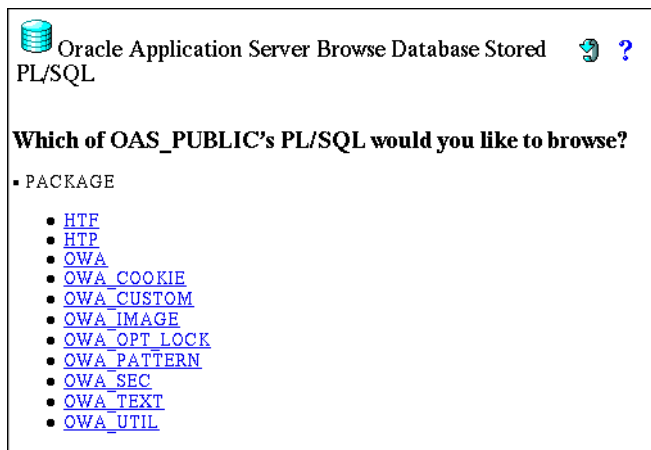
1. Select the Stored PL/SQL link to view PL/SQL functions, packages and procedures. A list of database users appears.

Figure 10–26 *Select User*



2. Click the user whose PL/SQL you would like to browse. A list of functions, packages, and procedures for that user are displayed.

Figure 10–27 *Stored PL/SQL*






3. Select the function, package, or procedure you would like to view.

The Database Browser utility displays the source code for the specified PL/SQL.

Database Users

Click Database Users to display a list of users and related information as shown in [Figure 10–28](#). A list of users appears along with each user's default tablespace, temporary tablespace, profile, and creation date.




Figure 10–28 Browse database users
 Oracle Application Server Browse Database Users  

Database user information:

Username	Default TS	Temporary TS	Profile	Created
DAVE	SYSTEM	SYSTEM	DEFAULT	28-JAN-99
DBSNMP	SYSTEM	SYSTEM	DEFAULT	01-DEC-97
OAS_PUBLIC	SYSTEM	SYSTEM	DEFAULT	28-JAN-99
SCOTT	TOOLS	TEMP	DEFAULT	28-JAN-99
SMARING	SYSTEM	SYSTEM	DEFAULT	08-FEB-99
SYS	SYSTEM	TEMP	DEFAULT	01-DEC-97
SYSTEM	TOOLS	TEMP	DEFAULT	01-DEC-97
TRACESVR	SYSTEM	SYSTEM	DEFAULT	01-DEC-97
WEBSYS	SYSTEM	SYSTEM	DEFAULT	28-JAN-99

Database Sessions

Click Database Sessions to display a list of current sessions connected to your database. A list of users appears, along with each user's associated operating system username, status, server, machine, and program. [Figure 10–29](#) shows an example of the information displayed.

Figure 10–29 Browse database sessions
 Oracle Application Server Browse Database Sessions  


Current sessions connect to your database:

Username	OS Username	Status	Server	Machine	Program
DAVE	dstahler	ACTIVE	DEDICATED	dstahler-sun	wrks@dstahler-sun (TNS V1-V3)


Rollback Segments

Click Rollback Segments to display rollback segment space allocation. A list of rollback segments appears with their owners, tablespace names, bytes, initial, next, extents, and maximum extents. [Figure 10–30](#) shows an example of the information displayed.

Figure 10–30 Browse database rollback segments



Oracle Application Server Browse Rollback Segments

 ?


Rollback segment space allocation:

Owner	Rollback Segment	TS Name	Bytes	Initial	Next	Extents	Max Extents
SYS	R01	RBS	931840	131072	131072	7	121
SYS	R02	RBS	931840	131072	131072	7	121
SYS	R03	RBS	798720	131072	131072	6	121
SYS	R04	RBS	931840	131072	131072	7	121
SYS	SYSTEM	SYSTEM	409600	51200	51200	8	121


Database Files

Click Database Files to view files, redo log files, and control files for your database, as shown in [Figure 10–31](#).

Figure 10–31 Browse database files



Oracle Application Server Browse Database Files

 ?

Database files for your database:

Tablespace Name	File Name	Bytes	Blocks
RBS	/private1/db8/point1/oradata/orcl/rbs01.dbf	15728640	7680
SYSTEM	/private1/db8/point1/oradata/orcl/system01.dbf	83886080	40960
TEMP	/private1/db8/point1/oradata/orcl/temp01.dbf	1048576	512
TOOLS	/private1/db8/point1/oradata/orcl/tools01.dbf	26214400	12800
USERS	/private1/db8/point1/oradata/orcl/users01.dbf	1048576	512

Redo log files for your database:

Thread#	Group#	Sequence#	Bytes	File Name	Group Status	Archived?	File Status
1	1	67	512000	/private1/db8/point3/oradata/orcl/redo01.log	INACTIVE	NO	
1	2	68	512000	/private1/db8/point2/oradata/orcl/redo02.log	INACTIVE	NO	
1	3	69	512000	/private1/db8/point1/oradata/orcl/redo03.log	CURRENT	NO	

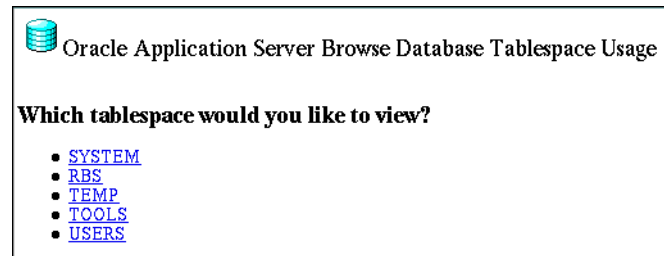
Control files for your database:

NAME	STATUS
/private1/db8/point1/oradata/orcl/control01.ctl	
/private1/db8/point2/oradata/orcl/control02.ctl	
/private1/db8/point3/oradata/orcl/control03.ctl	

Tablespace Space Usage

1. Click **Tablespace Space Usage** to browse database tablespace space usage. A list of tablespaces is displayed as in [Figure 10-32](#).

Figure 10-32 List of tablespaces



2. Click the name of the tablespace for which you want information. The result is displayed as shown in [Figure 10-33](#).

Figure 10-33 Browse tablespace usage

Oracle Application Server Browse Tablespace SYSTEM

Space usage for tablespace SYSTEM:

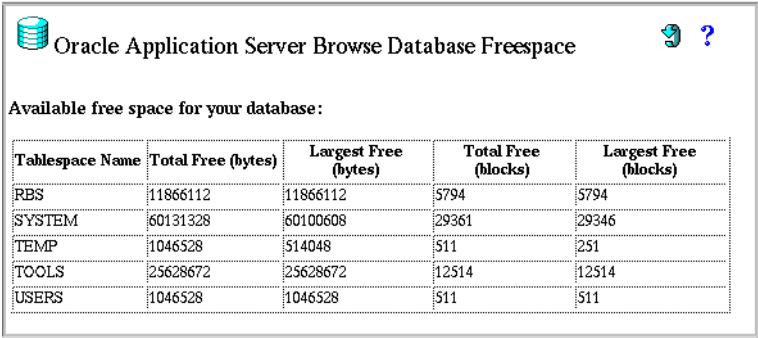
Owner	Segment Name	Segment Type	Extent ID	Starting Block	Blocks
SYS	SYSTEM	ROLLBACK	0	2	25
SYS	SYSTEM	ROLLBACK	1	27	25
SYS	C_OBJ#	CLUSTER	0	52	65
SYS	I_OBJ#	INDEX	0	117	5
SYS	C_TS#	CLUSTER	0	122	5
SYS	I_TS#	INDEX	0	127	5
SYS	C_FILE#_BLOCK#	CLUSTER	0	132	10
SYS	I_FILE#_BLOCK#	INDEX	0	142	5
SYS	C_USER#	CLUSTER	0	147	5
SYS	I_USER#	INDEX	0	152	5
SYS	UNDO\$	TABLE	0	157	5
SYS	FILE\$	TABLE	0	162	5

Tablespace Free Space Summary

Click **Tablespace Free Space Summary** to display the available space for your database. For each tablespace, the following information is shown (See [Figure 10-34](#)).

- total number of free bytes
- largest number of free bytes
- total number of free blocks
- largest number of free blocks

Figure 10–34 Tablespace free space summary



Troubleshooting

This section describes problems you may encounter when installing, configuring, or using Oracle Application Server Logging, and provides suggestions for fixing those problems.

Cause: Error messages do not show up in the Oracle Application Server error log immediately.

Action: Check to make sure the batch logging is turned off. Turning batch logging off will ensure that each message is sent directly to the logger. This can degrade system performance.

Cause: Error log file rolls over before reaching specified file size.

Action: If Archive Rollover has been set to Daily, Weekly or Monthly, this setting will override the Archiving File Size setting, causing the file to roll over before the specified file size is reached.

Cause: An error is generated while installing the PL/SQL package or using Log Analyzer.

Action: Check that the database is up and running.

Action: Check that the database user has the correct permissions.

Cause: User cannot access the OAS Admin Utility.

Action: Make sure that the admin listener is running.

Enabling Transactions

The Oracle Application Server transaction service provides the services necessary for applications and databases to become part of a transaction. Transactions allow the developer to combine specific operations together that the user sees as a single action. A transaction links together multiple operations and resource changes (database updates, file changes) into a single unit. These events or resources changes are distributed in some fashion, whether across multiple applications or multiple databases. Any transactions that occur within one database and within a single application can be taken care of through normal SQL programming.

Note: This feature is available only in the Enterprise Edition of Oracle Application Server.

Transaction theory, DTP, TX, and CORBA transaction protocols (OTS/JTS) are industry standards. As such, many books exist describing these standards. Therefore, this document only briefly explains these concepts.

Contents

- [Overview of Transactions](#)
- [Components Involved in the Transaction](#)
- [Configuring Your Transactional Environment](#)
- [Managing Your OTS Environment](#)

Overview of Transactions

- [What is a Transaction?](#)
- [When to use the Transaction Service](#)
- [Types of Transactions](#)

What is a Transaction?

A transaction links together multiple operations and resource changes into a single unit. All operations and resource changes that occur within the transaction all fail or succeed as a unit. For example, imagine a customer accessing a travel Web site who wishes to book a flight, a car, and a hotel. To the user, all of the bookings should occur within a single transaction. To the developer, all three bookings are three separate operations: one to the airline database, one to the car rental database, and one to the hotel database. These operations can occur concurrently or consecutively. When the user decides to accept all reservations, the application requests each database to commit the transaction's resources. The application treats each of the database requests as a separate operation. A transactional approach to this procedure enables the developer to implement all three operations as part of a single unit of work.

Note: This example used three databases; however, a transaction does not necessarily involve databases.

A transaction is a collection of operations that exist as a unit. The transaction service begins, commits, rolls back, and keeps track of operations that are within the transaction scope. In the above case, a full travel booking is considered a transaction. A transaction normally requires updates to more than one database. All updates to all included databases must succeed for the transaction to be considered successful. If any operation or database request fails within the transaction, the whole transaction is unsuccessful. In the case of an unsuccessful operation, all changes made for this transaction in all included databases are not saved, but are rolled back. If it is successful, changes to all databases involved are committed. A transaction guarantees that everything within the transaction either succeeds or fails.

An Oracle Application Server transaction can exist in a distributed environment. All participants (applications, objects, and databases) can exist on the same or separate nodes. The transaction provides synchronization across all participants of a transaction across all node boundaries.

Note: Oracle Application Server distributed transactions does not support nested transactions. DDL SQL statements like CREATE and ALTER commands cannot be executed with a distributed transaction. The commands perform an implicit transaction that acts like nested transaction.

When to use the Transaction Service

You can use the transaction service when you need to group database operations into a single unit, and when those operations span multiple requests or applications or databases.

If you need transactions, but the operations are within a single request, a single database, or a single application, you can simply use local transactions through SQL or transactions demarcated by the database API that you are using.

Types of Transactions

The transaction service enables you to demarcate transactions through APIs or URLs. Thus, there are two types of transactions:

- Programmatic transactions—defining transactions through either the Java Transaction Service (JTS) or TX APIs

In programmatic transactions, you use JTS or TX APIs to begin, roll back, or commit transactions. The particular function or method that you call depends on the programming language and transaction model.

- Declarative transactions—defining transactions through either URLs or a deployment descriptor

In declarative transactions, you associate URLs with begin, rollback, and commit operations. You also associate URLs with database operations that fall within the scope of the transaction.

Note: Within one transaction, you should use the same type; you should begin and end the transaction either programmatically or declaratively.

In addition to different ways of defining your transaction, there are several options for how to connect to the databases involved in the transaction. Each application type supports the transaction service differently both in terms of the type of transac-

tion definition, and what protocol to use when accessing databases. The following table lists these options for each application.

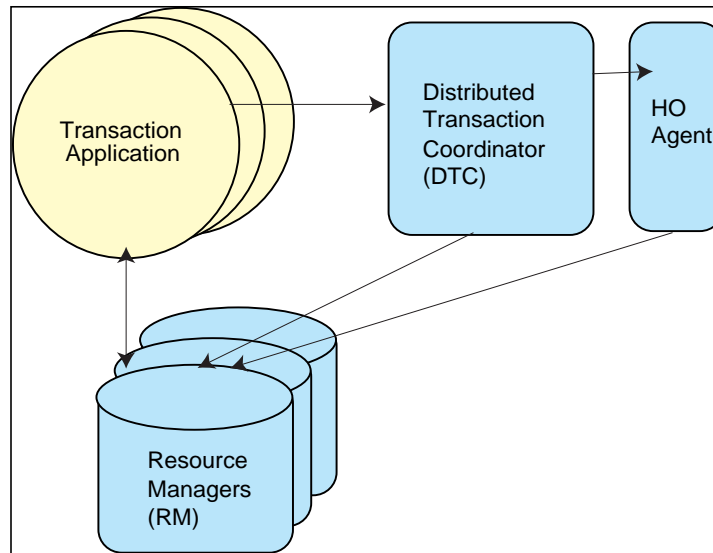
Table 11–1 Transaction service in applications

Application	Transaction Demarcation	Database Access
PL/SQL application	Declarative through URL	Application
JWeb application	Programmatic using JTS	JTS JDBC or pl2java
C application	Supports either Programmatic using TX or Declarative through URI	OCI or WDBC
LiveHTML application	Uses transaction tags, see the LiveHTML developer guide. No configuration needed.	ICX to PL/SQL application
Perl application	not supported	not supported
JCORBA and EJB objects	Programmatic through User Transaction Interface or Declarative using deployment descriptors.	JTS JDBC

Components Involved in the Transaction

OAS transaction service coordinates and manages each transaction and places no constraints on the number of objects involved or how the application is distributed across a network. OAS transaction service binds the objects involved together as a single entity and tracks the participants in a transaction. Once the transaction completes, Oracle Application Server ensures that all changes to all databases involved in the transaction are either committed or rolled back.

[Figure 11–1](#) shows the components involved in a transaction application.

Figure 11-1 Transaction component diagram

The following entities may be involved in your transaction, which requires that each entity is configured appropriately. The configuration requirements for these entities are described in on page 11-6.

Transaction Application

The application is responsible for starting a transaction, for connecting and disconnecting from all RMs, and for deciding whether to rollback or commit the transaction's database changes. The transactional functionality can exist either in a single application or across several applications. For programmatic transactions, if the functionality is spread across several applications, the transaction can be committed only in the application where the transaction was started.

Distributed Transaction Coordinator (DTC)

The DTC coordinates the operations necessary for committing changes in the databases, also known as Resource Managers (RMs). In this release, the DTC must be an Oracle8 database. The RM and DTC entities can be served by the same Oracle8 database.

If any errors occur during the commit process, all changes are guaranteed to be rolled back.

To use a DTC on a remote node, the remote node needs a listener component. Install this component when adding the remote node. See ["Adding a New Node"](#) on page 2-7.

Resource Manager (RM)

The application makes its data changes to an RM, which is an Oracle7 or Oracle8 database. Multiple RMs can exist in an environment to serve requests. Committing changes to multiple RMs is coordinated by the DTC. If there are multiple RMs, each RM does not need to be the same Oracle database version as all of the other RMs.

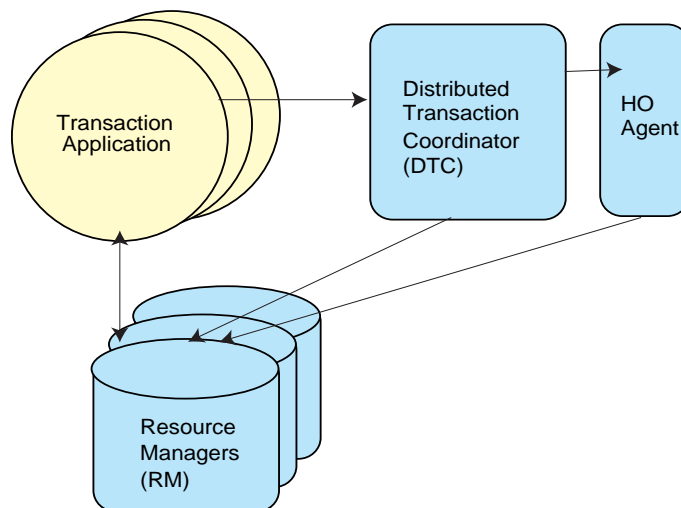
Heterogeneous OTS (HO) Agent

The HO Agent is an intermediary between an Oracle8 database and an Oracle7 database. If your RM database is an Oracle7 database, you need an HO Agent to mediate between your Oracle8 DTC and Oracle7 RM.

Configuring Your Transactional Environment

This section describes the configuration for each transaction component. [Figure 11-2](#) shows the entities that must be configured.

Figure 11-2 Transaction component diagram



You may configure all entities through the Oracle Application Server Manager.

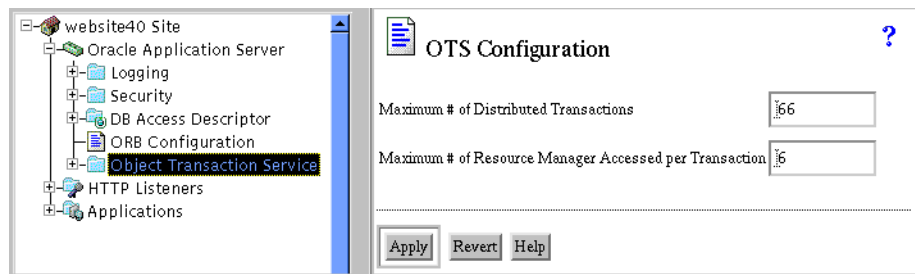
- RM - Configure a transactional DAD. A DAD is an access descriptor to a database. See [Chapter 6, "Database Access Descriptors \(DADs\)"](#) on how to configure the DAD. See on page 11-8 for how to make this DAD transactional.
- DTC - Configure the DTC so it can monitor multiple DADs for your transaction. See on page 11-12 for more information.
- HO Agent - Configure the HO Agent only if your DTC is Oracle8 and one of your transactional DADs is Oracle7. See on page 11-18 for more information.
- Transaction Application - Configure the transaction parameters within the application. See on page 11-20 for more information.

Object Transaction Service

Before configuring any of your entities, you may change any of the transaction tuning parameters on the Object Transaction Service (OTS) form.

1. Click Object Transaction Service to display the OTS Configuration form in the right frame (see [Figure 11-3](#)).

Figure 11-3 OTS Configuration form



2. The following describes the form's fields:
 - Maximum # of Distributed Transactions:
This value is the number of allowed transactions to be monitored by the Oracle Application Server transaction service and DTC. The maximum is set for performance reasons.
 - Maximum # of Resource Managers Accessed per Transaction:
This value is the maximum number of RMs (transactional DADs) for each transaction to enforce performance for this transaction.

3. Click Apply. Your Oracle Application Server transaction tuning configuration is now complete.

Configuring the DAD/Resource Manager

The RM is equivalent to a DAD that is configured to be transactional. Configuring the RM is a three-phase task including the following:

1. Configuring the database that will be used as the RM. This document does not discuss basic database configuration, as that topic is covered within your database's documentation set.
 - a. Each RM can be an Oracle7 or Oracle8 database. Configure the appropriate database using the Oracle database documentation.

Note: Some of the parameters from the database configuration are requested within the Oracle Application Server DAD configuration. Keep a note of the following configuration parameters for the Oracle Application Server DAD configuration: `db_name`, `db_domain`, global database name and `ORACLE_SID`. The combination of the `db_name` and `db_domain` form the global database name. If these are not defined in the database's `INITxxx.ORA` file, determine the value for the global database name by invoking "select * from global_name" within `svrmgrl`.

- b. Each database configuration requires network configuration. The Oracle databases use either SQL*Net or Net8 for their database connections. All connections are defined through a connect string. Each client that connects to the database should have a connect string defined within the client's `TNSNAMES.ORA`. For Oracle Application Server, this means that each node, primary or remote, will have a `TNSNAMES.ORA` file that should have a connect string for each RM database destination.

Note: If your DTC will exist on a different Oracle Home than the Oracle Application Server primary node, the connect string for the destination RM should also be copied into the DTC's `TNSNAMES.ORA`. The DTC also acts as a client to the RM and needs to be able to connect from its own `TNSNAMES.ORA` file.

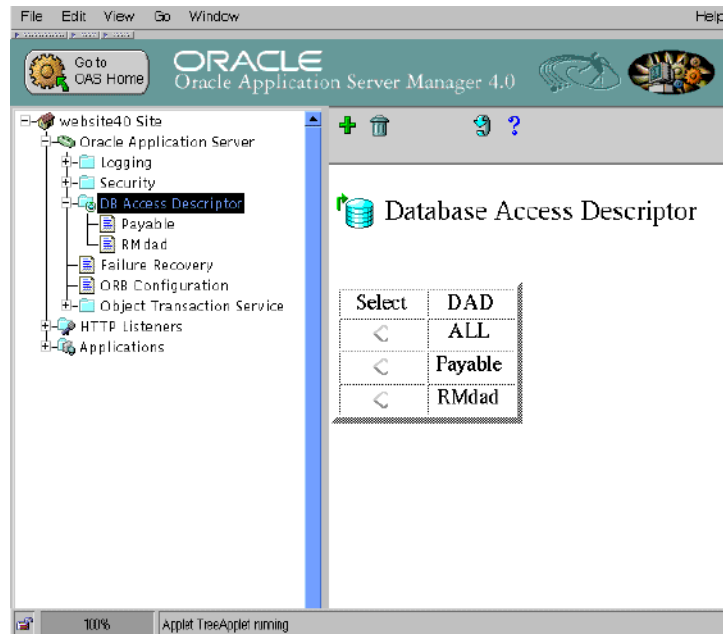
2. Adding DAD configuration to the existing database's configuration through the Oracle Application Server Manager. When creating your DAD for a transac-

tional application, you must configure both the ORACLE_SID and the connect string. See [Chapter 6, "Database Access Descriptors \(DADs\)"](#) for specific DAD configuration information.


Note: It is assumed that the RM database exists in the same Oracle Home as the Oracle Application Server primary node. If not, give the OAS Manager permission to change the RM database files, as well as to start and stop the database.

3. Adding transaction configuration to the existing DAD configuration through the Oracle Application Server Manager. This step is documented in this section.

Figure 11–4 Database Access Descriptor form

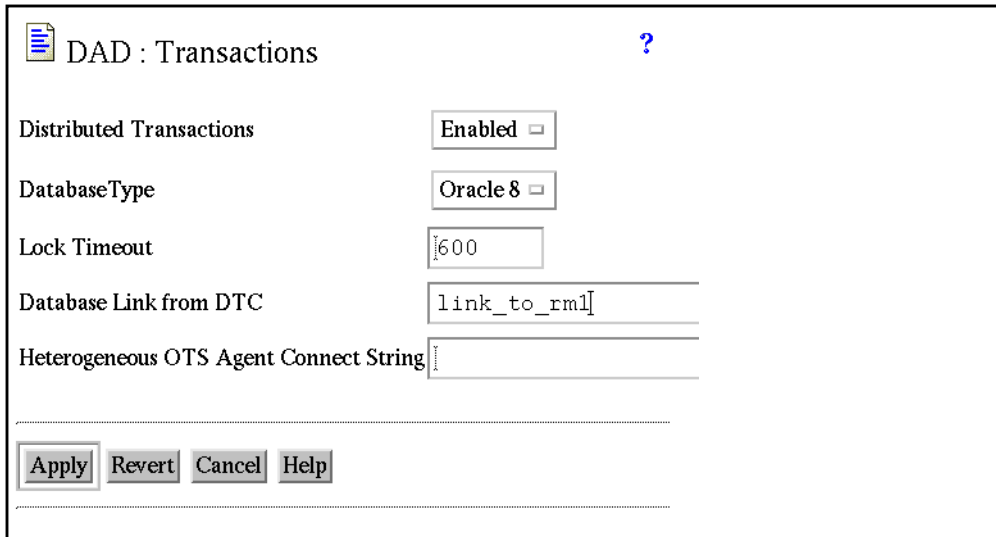


1. Click “DB Access Descriptor (DAD)” to display the Database Access Descriptor form (see [Figure 11–4](#)). This form shows all of the existing configured DADs.
2. Choose a DAD and configure it to be transactional:

- a. If the DAD is already configured, click the  next to the desired DAD. Click on the "Transactions" button.
- b. If the DAD is not configured, configure the DAD as described in [Chapter 6, "Database Access Descriptors \(DADs\)"](#). At the "Success" form, click on the "Transaction Enable this DAD Now" to configure the DAD as a RM.

This action displays the DAD Transactions form (see [Figure 11-5](#)).

Figure 11-5 DAD: Transactions form



DAD : Transactions ?

Distributed Transactions Enabled ▾

DatabaseType Oracle 8 ▾

Lock Timeout 600

Database Link from DTC link_to_rm1

Heterogeneous OTS Agent Connect String

Apply Revert Cancel Help

3. In the DAD: Transactions form:
 - Distributed Transactions enable/disable pulldown: Select "Enabled" to make the DAD transactional. Select "Disabled" to turn off transactions within the DAD.
 - Database Type: Choose either Oracle7 or Oracle8 based on what database you installed.

Note: If you chose Oracle7, you must configure the Heterogeneous OTS Agent Connect String. The HO Agent acts as an intermediary between the Oracle7 RM database and the Oracle8 DTC database. See on page 11-18 for specific instructions on configuring the HO Agent.

- **Lock Timeout:** If two requests are in contention for the RMs resources, the second request waits for the first to finish before its request is handled. This may produce a performance or deadlock situation. So, this value is the total time (in seconds) the second request is allowed to wait before it is rejected.
- **Database Link from DTC:** This can be any string that uniquely identifies the database link from the DTC to this RM. The DTC will use this name to identify the database link that the DTC creates. The following lists the guidelines in choosing this name:
 1. The string must be a unique name.
 2. Normally, you should use the global database name for the RM. Except in one situation, it is guaranteed to be unique. In the case that the DTC and the RM are the same database, you cannot use the global database name because it would create a self-referencing link. In this instance, use any other unique name. An easily identified unique name would be to append the DTC name on to the global database name (global_dbname_dtc_name) for the database link.
 3. If you do not use the global database name, choose any other unique string. You must set "global_names=false" in the DTC's init.ora file. If the global_names parameter is set to true, this database link is expected to be the RM's global database name.

The global database name is the combination of the database name and domain specified in the Oracle database initialization file (INITxxx.ORA). These names must be the same as what is specified within the INITxxx.ORA file and must conform to database naming rules.

- * **Database name:** the database name must be 8 characters or less. In addition, the database name cannot start with a number.
- * **Domain name:** the domain name recognizes where the database resides within a named environment. An example of a domain name is `us.oracle.com`. The domain name should not include the host name of where the database resides.
- **Heterogeneous OTS Agent Connect String:** This field is required if your RM is an Oracle7 database. If your RM is Oracle8, leave this field blank.

A Heterogeneous OTS (HO) Agent is necessary to interact between an Oracle8 DTC and an Oracle7 RM. If your RM is an Oracle7 database, you must configure an HO Agent to interact between them. This configuration field requires the HO Agent's listener service name configured in the DTC

Oracle8 LISTENER.ORA file. In [Example 11-1](#), the **ots_ho** service name is the value that you would supply in this field. See [page 11-18](#) for more information.

4. Click Apply. The RM, as a transactional DAD, is now configured.

Once the DAD is configured as transactional, it appears on the DTC list as one of the RMs available for the DTC to coordinate.

Note: If the DTC that requires this RM currently exists and is running, you must reload the OAS Manager for the new RM to appear on its lists.

Configuring the Distributed Transaction Coordinator

Configuring the DTC is a three-phase task:

Note: You must add the DTC only after all relevant transactional DADs have been added. If you add another transactional DAD after the DTC has been configured, you must remove the DTC and then add the DTC again to include the latest DAD.

1. Configure the Oracle8 database that is to be used as the DTC. An Oracle8 (Unix version 8.0.4 or later; Windows NT version 8.0.5.1 or later) database must be configured to be your DTC. In addition, the DTC database must exist on the same platform as the Oracle Application Server primary node.

Comment out the `open_links` parameter in the DTC's `init<SID>.ora` file. Oracle Application Server DTC configuration will include a configuration for the `open_links` parameter. If `open_links` is configured within the `init<SID>.ora` file, it overwrites the Oracle Application Server DTC configuration. Comment out the `open_links` parameter within `init<SID>.ora` to allow the `ots_<SID>.ora` configuration to take effect. If you do not do this, you may receive the following error message:

OWS-29046: "DTC doesn't have desired parameters, check your `initSID.ora`."

Note: If you do not comment out the `open_links` parameter within the `init<SID>.ora` file and you do receive this message, you must comment out the parameter, delete the Oracle Application Server DTC, and then add the DTC again.

See the *Oracle8 Administrator's Guide* for basic database configuration, as that topic is extensive and covered within your database's documentation set.

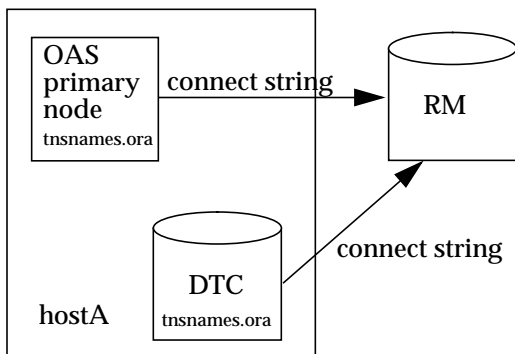
2. Configure SQL*NET V2 (Oracle 7 database) or Net8 (Oracle 8 database) connect strings from the DTC to the RMs. The Oracle Application Server primary node must be given access to the RM by configuring the RM's connect string into the Oracle Application Server TNSNAMES.ORA file.

Note: It is important that you configure the connect string in accordance with the TNSNAMES.ORA file. If the TNSNAMES.ORA file defines a default domain, all aliases will be appended with the domain. If the default domain is not defined, you need to configure the alias with the domain name within the TNSNAMES.ORA file as well as within Oracle Application Server configuration. An easy rule to follow is to always configure the alias exactly as it is configured within the TNSNAMES.ORA file. For Windows NT, you must always configure the alias with the domain name.

Since the DTC and the Oracle Application Server primary node can exist within separate Oracle Home directories, you may also have to modify the DTC's TNSNAMES.ORA file.

- If your DTC exists on the same Oracle Home as the Oracle Application Server primary node, the DTC uses the same TNSNAMES.ORA file as the Oracle Application Server primary node; thus, if you added the RM's connect string into the TNSNAMES.ORA file, the DTC will access the correct address.
- If your DTC exists on a different Oracle Home than the Oracle Application Server primary node, as shown in the following diagram, the connect string for the destination RM should also be copied into the DTC's TNSNAMES.ORA.

Figure 11–6 DTC in a different Oracle home



Start the Net8 listener that listens for the DTC. If you change the Net8 configuration files (TNSNAMES.ORA or LISTENER.ORA), you must restart the Net8 listener.

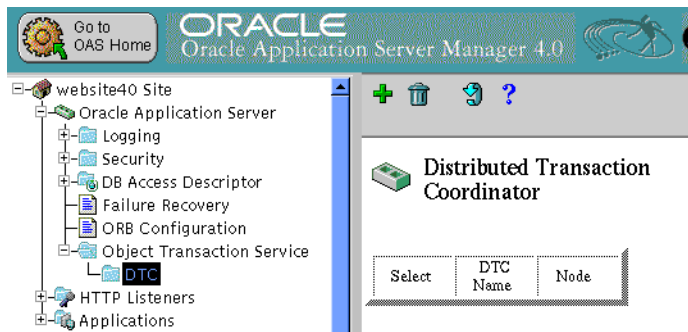
3. Configure the DTC through the Oracle Application Server Manager.

Note: Before creating a DTC, make sure that either the database is shutdown or all client connections to the database are closed.

Configuring the DTC

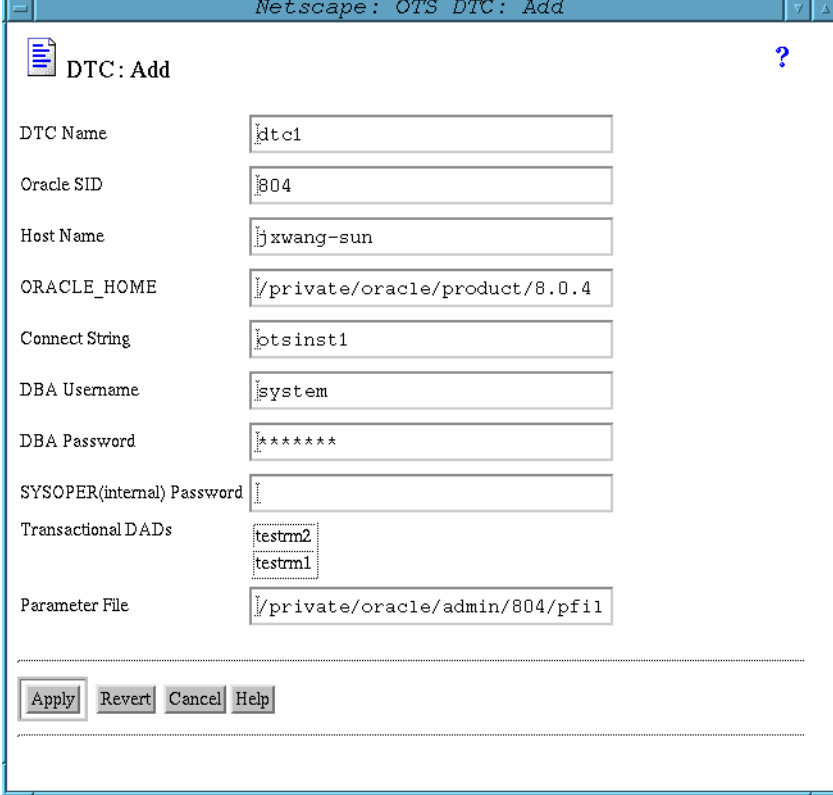
1. Click the next to "Object Transaction Service".
2. Expand the Object Transaction Service folder and click DTC to display the Distributed Transaction Coordinator form (see [Figure 11–7](#)).

Figure 11–7 Distributed Transaction Coordinator form



3. On the Distributed Transaction Coordinator form, click the  icon. This action displays the DTC: Add form (see [Figure 11-8](#)).

Figure 11-8 DTC: Add form



The screenshot shows a Netscape browser window titled "Netscape: OTS DTC: Add". The main content area is titled "DTC: Add" and contains the following fields:

- DTC Name:
- Oracle SID:
- Host Name:
- ORACLE_HOME:
- Connect String:
- DBA Username:
- DBA Password:
- SYSOPER(internal) Password:
- Transactional DADs:
- Parameter File:

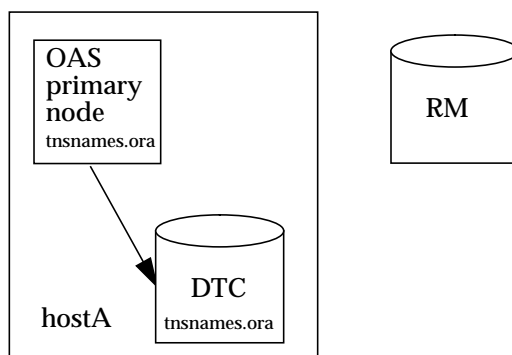
At the bottom of the form are four buttons: **Apply**, **Revert**, **Cancel**, and **Help**.

4. In the DTC: Add form:
 - DTC Name: An alias to identify your DTC.
 - ORACLE_SID: The ORACLE_SID of the Oracle8 database acting as the DTC. The administrator created this name during database configuration.
 - Host Name: The name of the host machine where the DTC resides. This name does not include the domain. For example, host **jxwang.us.ora-**

cle.com includes the hostname **jxwang** and the domain **us.oracle.com**. This field only requires **jxwang**.

Note: The host should be part of the Oracle Application Server site; if not, add this host to the web site before configuring the DTC.

- Oracle Home: The directory path where the Oracle8 database used as the DTC is located on the host machine.
- Connect String: This string is an alias that identifies the location and name of the DTC. As shown by the diagram below, this is the connect string that identifies the connection from the Oracle Application Server primary node to the DTC. This connect string alias has been configured within the Oracle Application Server Net8 TNSNAMES.ORA file, which may be located at \$ORACLE_HOME/network/admin, \$TNSNAMES.ORA, or another location specified by your SQLNET.ORA file.



Note: It is important that you configure the connect string exactly as it is configured in the TNSNAMES.ORA file. If the TNSNAMES.ORA file defines a default domain, all aliases will be appended with the domain. If the default domain is not defined, you need to configure the alias with the domain name within the TNSNAMES.ORA file as well as within Oracle Application Server configuration. For Windows NT, you must always configure the alias with the domain name within Oracle Application Server configuration and the TNSNAMES.ORA file.

- **DBA Name:** A database administrator name and password are required for accessing the Oracle8 database acting as the DTC. Enter the configured DBA name for this DTC.
- **DBA Password:** A database administrator name and password are required for accessing the Oracle8 database acting as the DTC. Enter the configured DBA password for this DTC.
- **SYSOPER Password:** This is the password that is used to allow the database instance to be started. Usually, a specific user, called the “internal” user, is only allowed to start or stop the database. This is the “internal” user’s password. This password is required on Windows NT, but is not required for Unix.
- **Transactional DADs:** A DAD can be used by transactional applications as a RM. If the DAD has been configured to be used as a RM, it appears in this list. If you expected a DAD to be listed that is not, go to the Database Access Descriptor configuration form and modify the DAD to be transactional.

Note: When you complete your DTC configuration, a database link will be created to each of these transactional DADs. If you add a new transactional DAD, this DTC must be deleted and added again.

- **Parameter File:** Input the location and name of the Oracle8 INITxxx.ORA parameter file. Because the DTC uses the Oracle8 database, the details of its configuration are necessary.
5. Click Apply. The DTC is now configured. In addition, database links with the same name as defined during RM configuration are created from the DTC to each RM/transactional DAD. You can verify the database links by typing the following command from Server Manager:

```
connect system/manager
select * from all_db_links;
```

Note: This new DTC can now coordinate transactions across all the RMs (transactional DADs) currently configured.

Configuring the HO Agent

If your RM (transactional DAD) database is an Oracle7 database, you need to configure the HO Agent that comes with the Oracle8 database (used as the DTC) to mediate between the DTC and the RM. Configuring an HO Agent requires modifying the DTC Oracle8 database's TNSNAMES.ORA file, LISTENER.ORA file, and adding the TNSNAMES.ORA service name to the RM configuration. The first two items are discussed here; the RM configuration item is discussed under the RM configuration section.

On a Unix platform, the HO Agent code is installed as part of the Oracle8 code base. On Windows NT, the HO Agent code is installed as part of the Oracle Application Server code base. After either type of installation, the HO Agent is installed as `hsots` (`hsots.exe` on Windows NT) in `$ORACLE_HOME/bin`. The HO Agent is considered as part of the Oracle8 database code base.

The following discussion assumes that you have already configured your Oracle8 database.

Modifying the TNSNAMES.ORA file

The Oracle8 database used as the DTC needs to have a database connect string created for the RM database. Since the RM is an Oracle7 database, the HO Agent needs to be added to the TNSNAMES.ORA file as the recipient of any requests directed from the DTC to the RM.

You must add an entry for the HO Agent in the TNSNAMES.ORA file with the following structure:

```
ho_agent_service_name = (DESCRIPTION=
  (ADDRESS =
    (PROTOCOL=TCP)
    (Host= Oracle8_hostname)
    (Port= Oracle8_portnumber)
  )
  (CONNECT_DATA =
    (SID = HO_SID)
  )
  (HS=)
)
```

where:

- `Oracle8_hostname`: the name of the host where the DTC Oracle8 database is installed.
- `Oracle8_portnumber`: the port number of the DTC Oracle8 database listener.

- **HO_SID**: the SID defined for the HO Agent in LISTENER.ORA. In the next section, "[Modifying the LISTENER.ORA file](#)", the HO Agent SID is defined as "ho_agent".

Note: The HO Agent service name is the connect string required in the DAD: Transactions Heterogeneous OTS Agent Connect String field.

Modifying the LISTENER.ORA file

You need to add a listener for the HO Agent in the Oracle8 LISTENER.ORA file.

```
SID_LIST_listener = (SID_LIST=
  (SID_DESC=(SID_NAME=rdms80)(ORACLE_HOME=/private/oracle))
  (SID_DESC=(SID_NAME=rdms70)(ORACLE_HOME=/private/ORA7))
  (SID_DESC=(SID_NAME=ho_agent)(ORACLE_HOME=/private/oracle)(PROGRAM=hsots))
```

where:

- The HO Agent points to the same \$ORACLE_HOME as the Oracle8 database.
- The HO Agent SID_NAME is the same name defined in the HO Agent's CONNECT DATA in TNSNAMES.ORA.
- The HO Agent listener adds the parameter of PROGRAM=hsots. This describes the executable to be started up when the RM is requested. This SID_NAME is the service name provided in the RM configuration.

Example 11-1 HO Agent configuration example

The following example shows how the TNSNAMES.ORA and LISTENER.ORA files are related to each other in their configuration of the HO Agent.

```
/*TNSNAMES.ORA file configures an HO Agent called test_ho*/
ots_ho =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (Host = jxwang-sun.us.oracle.com)
      (Port = 1521)
    )
    (CONNECT_DATA =
      (SID = test_ho)
    )
    (HS=)
  )
```

```
/*LISTENER.ORA file defines listener for HO Agent called test_ho*/
SID_LIST_ots_listener = (SID_LIST=
  (SID_DESC=(SID_NAME=804)(ORACLE_HOME=/private/oracle/product/8.0.4))
  (SID_DESC=(SID_NAME=test_ho)(ORACLE_HOME=/private/oracle/product/8.0.4)
    (PROGRAM=hsots))
)
```

Configuring the Transaction Application

Applications can use two types of transactions: programmatic or declarative. Each type is dependent on the application language.

Table 11–2 Transaction service in applications

Application	Transaction Demarcation
PL/SQL application	Declarative through URI
JWeb application	Programmatic using JTS
C application	Supports either Programmatic using TX or Declarative through URI
LiveHTML application	No configuration needed for its transactions
Perl application	Not supported
JCORBA and EJB objects	Programmatic through JTS or Declarative using deployment descriptors

Both transactional types require that you enable transactions and specify the transactional DADs to use within the transaction. In addition, declarative transactions require an additional step. The following sections describe the transaction configuration:

- [Configuring General Transactional Application Parameters](#)
- [Configuring Declarative Transaction Parameters](#)

For more information about developing transactions, see the Oracle Application Server *Developer’s Guide* for the application you are interested in.

Configuring General Transactional Application Parameters

To configure the general transaction parameters for both programmatic and declarative transactions, you must modify the Transactions form. The following lists the details:




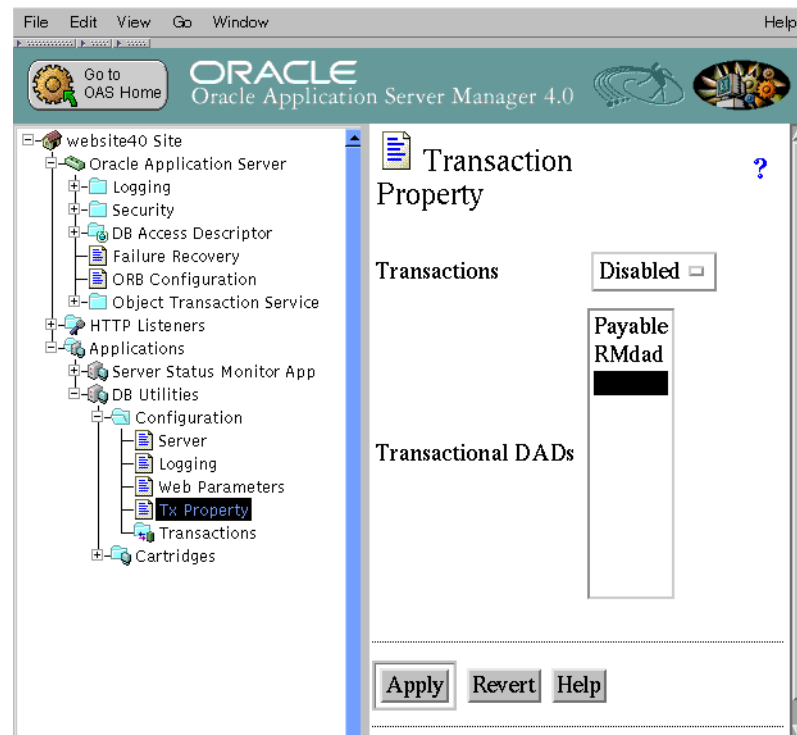
1. Click the  next to Applications to display the configured applications in the navigational tree.
2. Click the  next to the appropriate application listed below "Applications".
3. Click the  next to "Configuration" under the appropriate application.

Figure 11–9 Transaction Property form



4. Click Tx Property to display the Transaction Property form.
 - Transactions: Select "Enabled" to make the application transactional and "Disabled" to disable transactions within the application.

- Transactional DADs: Choose one or more RMs that your application will use.

Note: You cannot use transactional and non-transactional DADs in the same application. You need to separate your cartridges into separate applications based upon whether they use transactions; all cartridges within each application must either all be transactional or all non-transactional.

- Click Apply. The application is now installed and configured as transactional.

Configuring Declarative Transaction Parameters

Declarative transactions allow you to use transactions without writing any code. Transactional services provide the ability to start and end transactions, demarcate the transaction boundaries, and make the decision whether to commit or rollback each transaction. Declarative transactions exist so that the cartridge will invoke the transaction operations on behalf of your cartridge. You only need to either reference the configured URI (for PL/SQL or C) within your cartridge or invoke a deployed transactional object (for EJB or JCORBA). There are two types of declarative transactions:

- [Web Declarative Transactions](#) for PL/SQL and C cartridges
- [Deployment Declarative Transactions](#) for EJB and JCORBA cartridges.

You can define multiple transactions in the same application, where each transaction has its own set of virtual paths.

Web Declarative Transactions Web declarative transactions use URIs for all transaction operations. When your cartridge references the appropriate URI, the begin, commit, and rollback operations are executed for you by the cartridge runtime. The cartridge runtime executes the appropriate transaction call when the appropriately configured URI is requested.

Note: A URI is the generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is only one kind of URI.

For example, you create the a PL/SQL stored procedure within the transactional database and name it `begin`. The stored procedure can contain any type of PL/SQL statements that you consider appropriate to occur right after the transaction begins. You configure the PL/SQL stored procedure, `/hr/emp/begin`, as the Begin URI. When the client requests the `/hr/emp/begin` stored procedure, the PL/SQL cartridge executes the transactional `begin` for you and then invokes the `begin` stored procedure within the transactional database.

All operations that are to happen within the transaction scope—between the `begin` and the `commit` or `rollback`—are also URIs.

- | | |
|--------|--|
| PL/SQL | <ul style="list-style-type: none"> ■ Stored procedure—You can create and store PL/SQL procedures within the transactional database that will be executed after the cartridge executes the transactional protocols. ■ PL/SQL source files—You can create PL/SQL source files that exist in your directory, which will be executed after the cartridge executes the transactional protocols. |
| C | <p>The URI is just a dummy name. No actual file or code exists within a directory. However, you can execute code around the URI being referenced.</p> |

To configure the Web Declarative transaction, you must configure the virtual path and then configure the URI.

Configure the Virtual Path All URIs for this transaction must exist within the same virtual path of the application. For example, if the virtual path for the application is `/hr/emp`, you can set the virtual path for `Begin Transaction` as `/hr/emp/begin`. Virtual paths are configured under the cartridge configuration on the Virtual Path form.






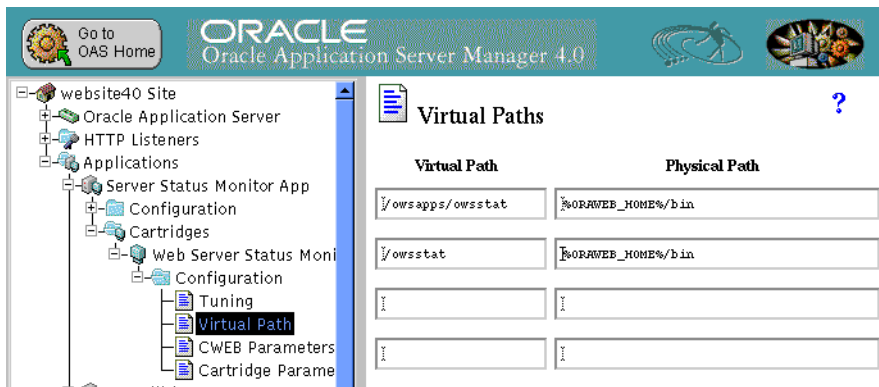
1. Click the  next to Applications to display the configured applications in the navigational tree.
2. Click the  next to the appropriate application listed below "Applications".
3. Click the  next to "Cartridges" under the appropriate application.
4. Click the  next to the appropriate cartridge listed below "Cartridges".
5. Click the  next to "Configuration" under the appropriate cartridge.
6. Select the "Virtual Path" to display the Virtual Paths form.

Figure 11–10 Virtual Path Form



Configure the following for each cartridge type:

PL/SQL

- **Stored procedure**—Enter a path string that is unique within Oracle Application Server to enable Oracle Application Server to identify that any incoming request with this string should be forwarded to the PL/SQL cartridge. This path does not need to exist. The "Physical Path" parameters are ignored for stored procedures.
- **PL/SQL source files**—Enter a path string that is unique within Oracle Application Server. Associate this path with the directory on your system where the PL/SQL source files exist.

C

Enter a path string that is unique within Oracle Application Server to enable Oracle Application Server to identify that any incoming request with this string should be forwarded to the C cartridge. This path does not need to exist. The "Physical Path" parameter is ignored.

7. Click Apply. Now you must configure the URIs.

Configure URI A URI is different for each cartridge.

Note: This form asks for URIs to the web declarative transactional pages. A URI is the generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is only one kind of URI.

Configure the following for each cartridge type:

- | | |
|--------|--|
| PL/SQL | <ul style="list-style-type: none"> ■ Stored procedure—configure the virtual path followed by the stored procedure name. The procedure should be stored within the transactional database. The cartridge runtime will execute the appropriate transactional command and then execute the stored procedure. ■ PL/SQL source files—Configure the virtual path followed by the PL/SQL source file. The PL/SQL source file must exist within the directory. The cartridge runtime will execute the appropriate transactional command and then execute the PL/SQL source file. |
| C | <p>Enter the virtual path followed by a dummy name. No C source or library files are necessary to exist with the dummy name. When the cartridge receives the URI with the dummy name, it executes the appropriate transactional command and returns.</p> |

To configure the URI, do the following:





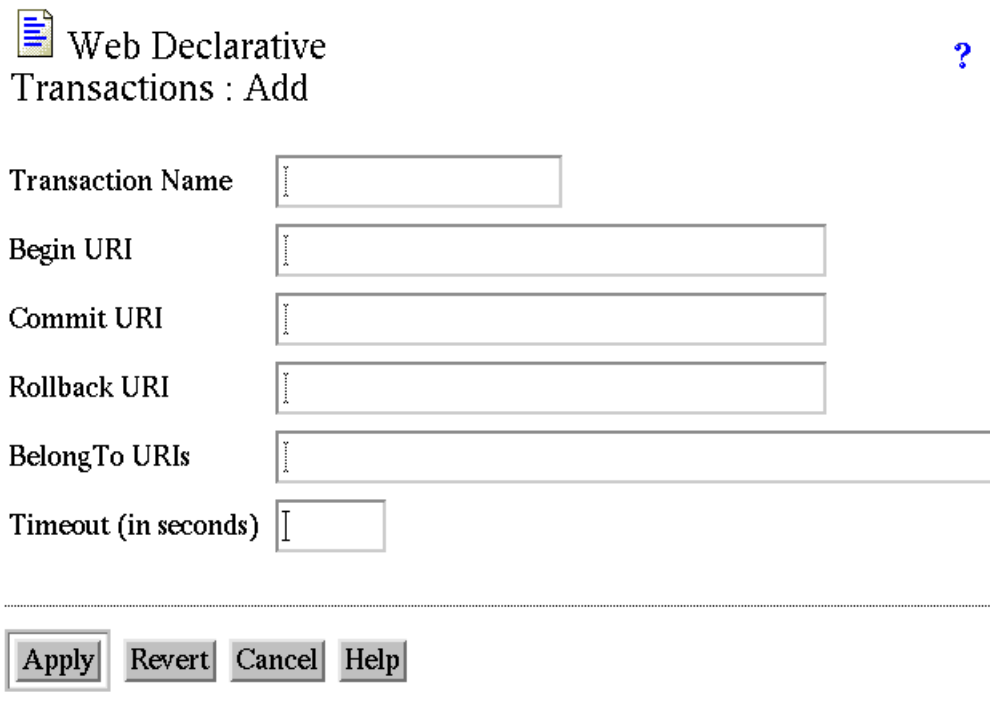


1. Click the  next to Applications to display the configured applications in the navigational tree.
2. Click the  next to the appropriate application listed below "Applications".
3. Click the  next to "Configuration" under the appropriate application.
4. Click Tx Property to display the Transaction Property form (see [Figure 11-9](#)).
 - Transactions: Select "Enabled" to make the application transactional and "Disabled" to disable transactions within the application.
 - Transactional DADs: Choose one or more RMs that your application will use.
 - Click Apply.
5. Click Transactions to display the Web Declarative Transactions form ([Figure 11-11](#)).
6. On the Web Declarative Transactions form, click the  icon. This action displays the Web Declarative Transactions: Add form (see [Figure 11-11](#)).

Figure 11–11 Web Declarative Transactions: Add form


 Web Declarative Transactions : Add 

Transaction Name

Begin URI

Commit URI

Rollback URI

BelongTo URIs

Timeout (in seconds)

7. In the Web Declarative Transactions: Add form:

- **Transaction Name** - A name that should be unique within the application to identify the transaction within any management displays. For example, `txn_shop`. Since your application can have multiple transaction, each is identified with this name.
- **Begin URI** - The URI to start the transaction. For example, `/plsql_dba/plsql/txn_shop.enter_url` where `plsql_dba` is the name, `txn_shop` is the package name, and `enter_url` is the procedure.
- **Commit URI** - The URI to commit an on-going transaction. For example, `/plsql_dba/plsql/txn_shop.commit_url`.
- **Rollback URI** - The URI to rollback an on-going transaction. For example, `/plsql_dba/plsql/txn_shop.rollback_url`.
- **BelongTo URIs** - The list of URIs that belong to the transaction. For example, `/plsql_dba/plsql/txn_shop.*`.

Between the begin and the commit or rollback URL, the user would invoke other URLs that call procedures to perform some action on the database. These procedures might or might not be within the transaction boundary. If the URI is within the transaction boundary, the actions performed by that procedure would be committed or rolled back when the transaction ends. If the URI is not within the transaction boundary, it is not affected by the transaction, and the Oracle Application Server treats it as a regular request (changes made by that URL are committed upon completion).

The `BelongTo` field in the Web Declarative Transactions form (Figure 11-11) lists the virtual paths that fall within the scope of the transaction. You may have multiple URIs defined within the `BelongTo` URI. Multiple virtual paths in the `BelongTo` URIs field can be defined in one of two ways:

- Multiple URIs are specified individually and are separated by the comma character. The virtual paths must use the application's virtual path as a base. For example, if the application's virtual path is `/hr/emp`, the `BelongTo` field can contain virtual paths such as `/hr/emp/updateEmp`, `/hr/emp/removeEmp`, and `/hr/emp/addEmp`.
 - Multiple URIs can be specified through a wildcard character, `'*'`, to denote multiple pages within the application. For example, `/hr/emp/emp.*`.
 - Timeout (in seconds) - The time elapsed (in seconds) before a transaction is terminated. For example, `90`. If the timeout occurs, the transaction is rolled back. The default is 10 minutes (600 seconds).
8. Click Apply. Your application is now transaction enabled. Restart all Oracle Application Server processes for the changes to take effect.

Deployment Declarative Transactions With EJB and JCORBA objects, you do not need to do any application or cartridge configuration through the GUI. The cartridge is configured to be declarative transactional within the Deployment Descriptor. Thus, when you invoke a method on an object that was deployed as transactional, the cartridge executes the begin. When the method completes, the cartridge either commits or rolls back the transaction, as appropriate. See the JCORBA and EJB Developer's Guide for instructions on adding the transaction properties into your Deployment Descriptor.


For configuring Deployment declarative transactions, you still must configure the transactional DAD and the DTC, as specified in the earlier steps.

Managing Your OTS Environment

This section describes how to start up and shut down your OTS environment.

Starting OTS Entities

Once everything is configured, you need to start up your OTS entities.

1. Start appropriate RMs. Start the databases that have been configured to be the RMs for your application. Databases can be started through several different methods. The most popular methods are through Oracle Enterprise Manager or through the command line tool: `svrmgrl`.
2. Start the DTC by starting the Oracle8 database and database listeners that act as the DTC.
3. Start Oracle Application Server 4.0. Start from the Oracle Application Server Manager. Click Oracle Application Server to display the Oracle Application Server Manager form. Select ALL and click  to start the Oracle Application Server components.

Stopping OTS Entities

To stop your OTS entities:


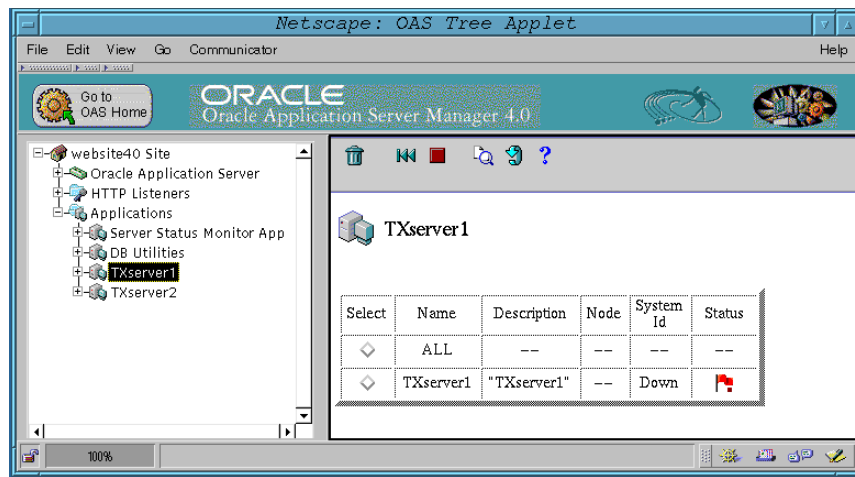
1. Stop the transaction application. You may use the Oracle Application Server Manager to stop any active applications.
 - Click the  next to Applications to display the configured applications in the navigational tree.
 - Click your configured application. In [Figure 11-12](#), the application name is `TXserver1`.
2. Stop appropriate RMs and DTCs. Shutdown the databases that have been configured to be the RMs or DTCs for your application. Databases are shutdown through several different methods. The most popular methods are through Oracle Enterprise Manager or through the command line tool: `svrmgrl`.

Figure 11-12 JCORBA Application form

- Select the application that you want to stop.
 - Click to stop the application.
3. Stop Oracle Application Server 4.0. From the Oracle Application Server Manager, click Oracle Application Server to display the main Oracle Application Server form. Select ALL in the form and click to stop the Oracle Application Server components.

Deleting and Browsing Transaction Entities

If you wish, you can delete your transactional configuration.

- [Deleting your Transactional Configuration for the Application](#)
- [Deleting a DTC](#)
- [Displaying an Existing DTC's Configuration](#)

Deleting your Transactional Configuration for the Application

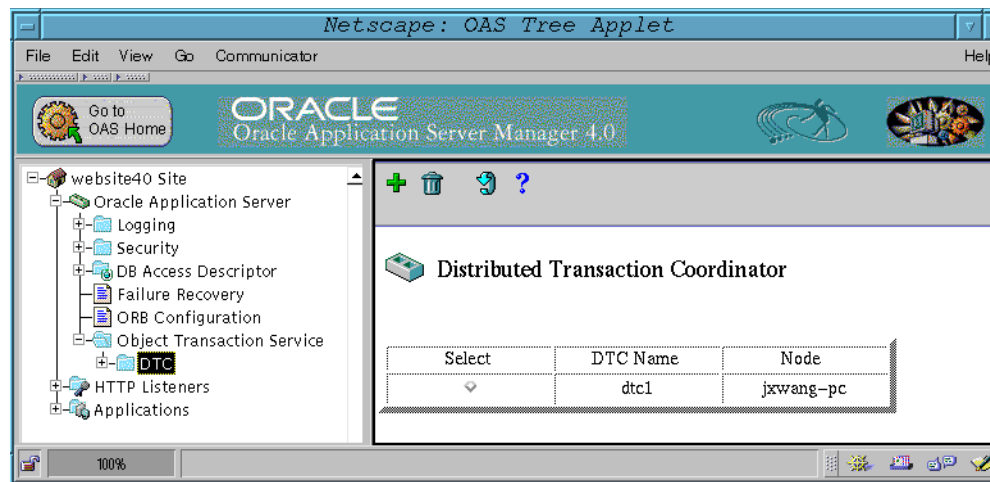
For any Programmatic transactions, choose the "Tx Property" form and disable Transactions.

For any Declarative transactions, click the next to the "Transactions" form, select the appropriate declarative application configuration, and delete the appropriate declarative configuration by choosing the .

Deleting a DTC

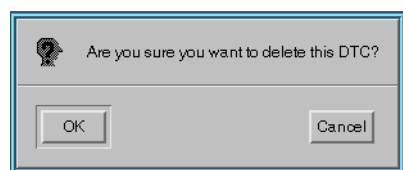
1. On the Distributed Transaction Coordinator form, click the DTC you want to delete (see [Figure 11-13](#)).

Figure 11-13 DTC Selection form



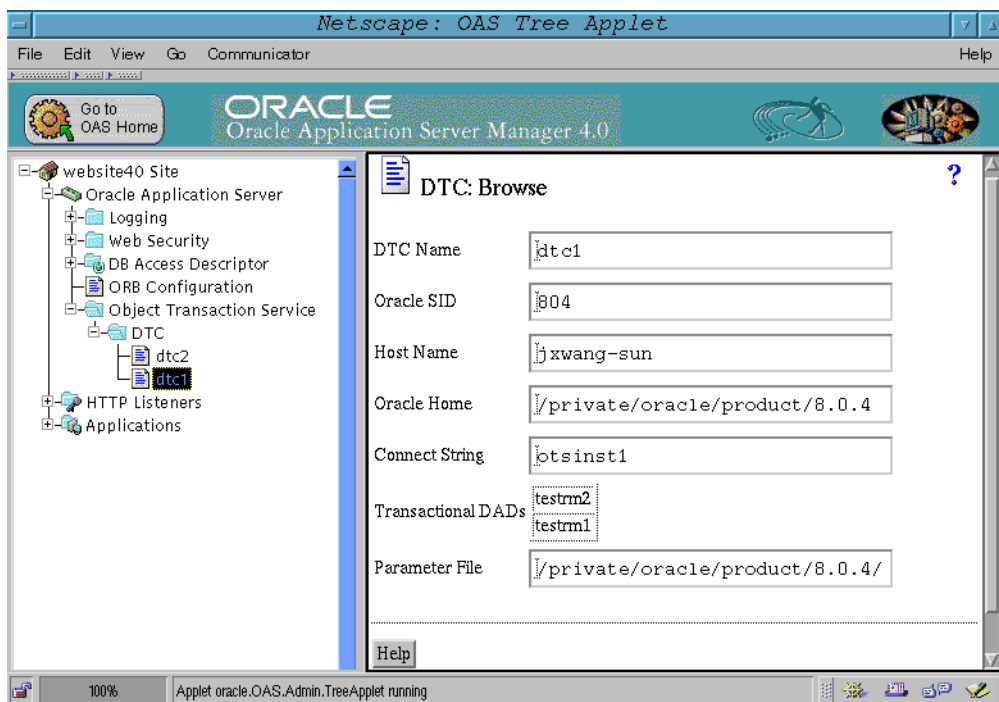
2. Once the DTC is selected, click the  icon. The Delete DTC Confirmation dialog box appears (see [Figure 11-14](#)). Click OK if you still want to delete the DTC.

Figure 11-14 DTC Deletion popup



Displaying an Existing DTC's Configuration

In the left frame, click the DTC you want to modify. This displays the DTC's current configuration in the right frame (see [Figure 11-15](#)).

Figure 11–15 DTC Display form

Failure Recovery for Oracle Application Server

An enterprise strength application server must tolerate a variety of failures and continue to function and serve user requests with minimal perceived downtime. The application server must be able to mask failures from end users and offer high service availability. Oracle Application Server uses a failure recovery system to recover automatically from failures, reduce system downtime, and achieve high availability. This chapter describes how failure recovery protocols function in Oracle Application Server.

Contents

- [Failure Recovery Components](#)
- [Failure Recovery Architecture](#)
- [Failure Recovery Protocols](#)

Note: On Windows NT, users should not use the Task Manager to terminate an Oracle Application Server process.

Failure Recovery Components

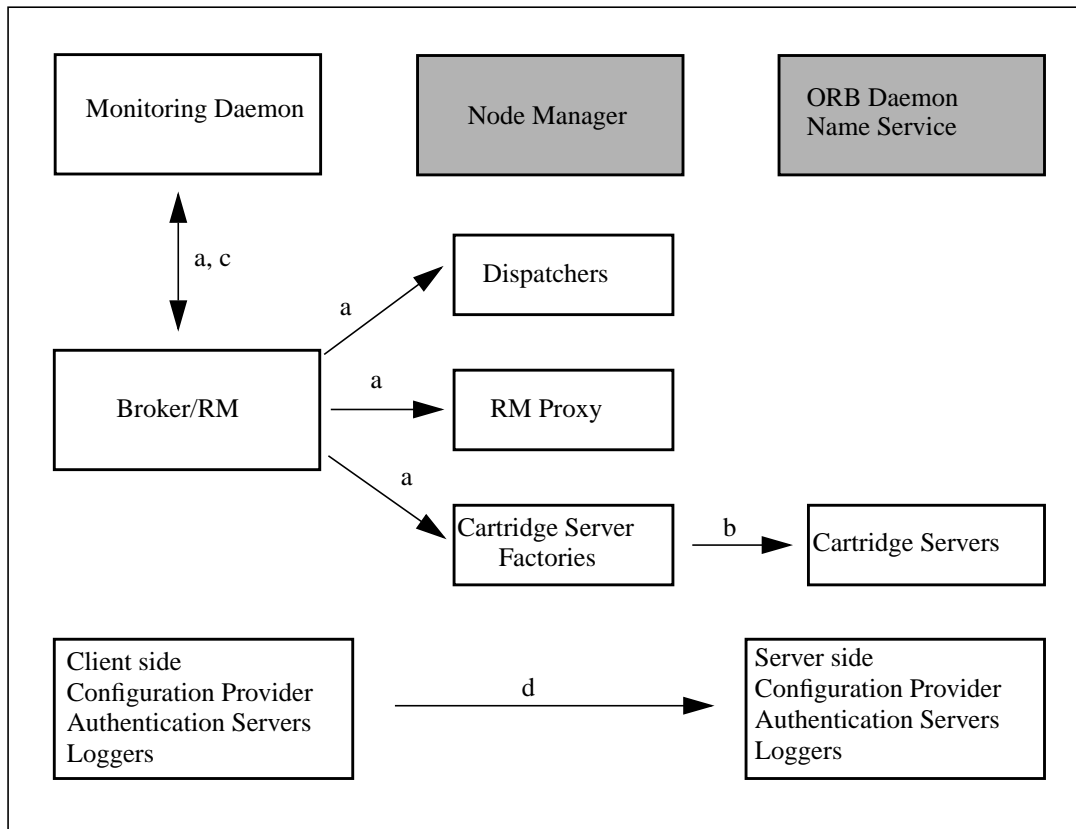
Failure Recovery allows you to enable or disable failure recovery for an individual component of Oracle Application Server. Valid components are:

- **Config Provider** — Reads configuration information from the **wrb.app** and supplies that information to any system component, either remote or local, that needs it. The Config Provider resides only on the primary node in a multi-node installation.

- **Authentication Server** — An object that encapsulates the authorization performed against cartridges. An authentication server consists of one authorization broker and several authorization providers.
- **Broker/RM** — The portion that evaluates and responds to requests. This component is also called the Resource Manager. There is one Resource Manager per site.
- **Logger** — Allows cartridges to write errors, warnings, or other useful messages to a central repository (a file system or a database).
- **Cartridge Server Factory** — A process that starts up cartridge servers. A cartridge server is a process that runs one instance of an application.
- **Dispatcher** — A dispatcher manages a pool of cartridge instances running on one or more nodes; each cartridge server runs one or more cartridge instances. There is one dispatcher associated with each listener on each node of a Web site.
- **Monitoring Daemon** — Allows you to check the status of Oracle Application Server processes and objects.
- **RM Proxy** — A resource manager proxy is responsible for obtaining JCO references on behalf of external clients.

Failure Recovery Architecture

The key pieces of the Oracle Application Server failure recovery architecture are shown in [Figure 12-1](#). The node manager and ORB Daemon/Name Service processes are shaded to indicate that they must be restarted manually if they fail. All other processes are equipped with automatic failure recovery.

Figure 12–1 Oracle Application Server failure recovery architecture

Oracle Application Server uses a distributed failure detection and recovery strategy. Rather than putting the burden on a single component to do the failure detection and recovery of all components, Oracle Application Server uses a distributed failure detection and recovery mechanism.

- a. The broker is responsible for monitoring and restarting dispatchers, RM Proxy, cartridge server factories, and the monitoring daemon.

Note: If a cartridge server process fails, it is recovered as soon as the RM Broker detects that it is down. This may take approximately five to ten minutes, depending on machine load.

Note: Failure recovery is not supported for Microsoft Internet Information Server 4.0 listeners.

- b. The cartridge server factories monitor the server processes they create and then tell the broker to clean up accordingly if they detect that a server process exited abnormally.
- c. The monitoring daemon periodically pings all the objects it retrieves from the Name Server. When it detects that the broker is down, it will stop the entire WRB and listener processes, and then restart them again.

In the Enterprise Edition, the monitoring daemon will bring up another broker if it detects that the broker is down.

- d. The client side of the configuration provider, authentication servers, and loggers each help monitor their corresponding servers, and request the broker to restart any servers that are down.

Failure Recovery Protocols

Failure Recovery of Individual Components

Oracle Application Server supports failure recovery for the following processes.

Process	Recovery Action
Monitoring Daemon	The monitoring daemon is restarted by the broker. The monitoring daemon gets each component's object reference from the Name Server which is persistent.

Process	Recovery Action
Broker/RM	<p>No broker state recovery occurs. When the monitoring daemon detects that the broker is down, it will stop all WRB and listener processes and restart them.</p> <p>Note: In the Enterprise Edition, broker state recovery occurs. The state of the broker is written to disk at one minute intervals. These files can be found in \$SORAWEB_ADMIN\$SORAWEB_SITES\wrb\data. In the Enterprise Edition, only the broker will be restarted when it is down.</p>
Dispatchers/Listeners	<p>When the broker detects failure of a dispatcher/listener, it deregisters the dispatcher and frees any cartridge instance object references cached at that listener. The broker then restarts a new listener process.</p>
RM Proxy	<p>RM Proxy is monitored by the broker. All outstanding client requests made at the time the process goes down are lost. Clients will have to make another new request again. Clients of RM Proxy should expect CORBA_EX_COM_FAILURE and CORBA_EX_INV_OBJREF exceptions thrown by the ORB if the RM Proxy exits abnormally.</p>
Cartridge Server Factories	<p>On detecting cartridge server factory failure, the broker starts a new cartridge server factory.</p> <p>One of the cartridge server factory's responsibilities is to monitor the cartridge server processes it creates. If a server process has exited abnormally, the cartridge server factory informs the broker to deregister the server.</p> <p>If the node where the cartridge server factory is running is entirely down, the broker will not be able to bring up another cartridge server factory there. In such a case, the system administrator may want to start up a cartridge server factory manually on some other node.</p>

Process	Recovery Action
Cartridge Servers	<p>The cartridge server factory informs the broker when it detects cartridge server failure. The broker removes all object references for all cartridge instances that exist within that cartridge server as well as notify the dispatchers that may have cached an object reference to that cartridge.</p> <p>In the case of a JCORBA cartridge server, since there is no way of telling the external CORBA client, the client will find out through an exception when the client tries to invoke a method on the cartridge instance object.</p> <p>The cartridge server will not be restarted. When a request of the type handled by the cartridge server is sent to the broker, the broker then brings up a new cartridge server.</p>
Configuration Providers	Client side of configuration provider calls the broker to restart configuration provider process.
Authentication Servers	Client side of authentication server calls the broker to restart authentication server process.
Loggers	Client side of logger calls broker to restart logger process.

Command Line Utilities

This appendix describes Oracle Application Server command-line utilities. These utilities can be found in the `$ORAWEB_HOME/bin` directory.

Note: Users familiar with 3.x versions of the Web server should note that the syntax of the `owsctl` command has changed for Oracle Application Server 4.x.

Contents

- [The `owsctl` utility](#) — used to stop, start, and monitor the status of the Web Request Broker (WRB), Object Request Broker (ORB), and listeners. You may also use the Oracle Application Server Manager to perform many of the `owsctl` functions described here.
 - [version](#)
 - [start](#)
 - [stop](#)
 - [status](#)
 - [reload](#)
 - [dump](#)
 - [monitor](#)
 - [getIOR](#)
- [The `owsstat` utility](#) — used to monitor the status of the Web Listener.

- [The oasnetconf utility](#) — allows you to change certain node properties on different machines.
- [The oaspasswd utility](#) — allows you to change Node Manager, ORB, and wallet manager passwords.
- [The oasmcastcfg utility](#) — allows you to display or change the Group Messaging Configuration set during installation.
- The wrcfreg utility — is used to register and deregister runtimes, applications and cartridges with Oracle Application Server.
Details for the wrcfreg Utility can be found in *Cartridge Management Framework*.
- The genreq utility — generates SSL certificate request text.
Details for the genreq utility can be found in the *Security Guide*.

The owscctl utility

The **owscctl** utility is a command line utility to administer Oracle Application Server. It starts, stops, reloads, and displays the status of the Oracle Web Listener, Oracle Web Status Monitor, WRB, ORB and Cartridge processes.

If you choose to use the command line utility to start Oracle Application Server, you should also use the command line utility to stop Oracle Application Server. Likewise, if you use the Oracle Application Server Manager to start Oracle Application Server, you should also use it to stop Oracle Application Server.

Note: Oracle strongly recommends that you use the Oracle Application Server Manager, a graphical tool for the configuration and management of Oracle Application Server, to administer your site.

To use **owscctl**, type:

```
owscctl
```

followed by the desired argument at your operating system's prompt.

On Unix, you must have the following environment variables set:

- **ORAWEB_HOME** — This is the absolute path where Oracle Application Server is installed, and should be **\$ORACLE_HOME/ows/4.0** for Release 4.0.8.

- **ORAWEB_SITE** — This is the site name for Oracle Application Server.
- **ORACLE_HOME** — This is the absolute path where you install Oracle products.

For a summary of **owsctl** syntax, type:

```
owsctl -h
```

The **owsctl** utility accepts the arguments outlined below.

Version Number

To display the version number of Oracle Application Server, type the following at the command line:

```
owsctl -V
```

Note: The command is case sensitive. Be sure to specify a capital V as the argument.

version

Purpose

To display the version number of web cartridges.

Syntax

```
owsctl version [ -T ] [ -c Web_Cartridge_Name ]
```

Arguments

-T	“Talkative” — this argument puts owsctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify -T if you want owsctl to operate in verbose mode.
-c	Prints the version of the web cartridge you specify.

start

Purpose

Used to start processes from the command line.

Note: You cannot start an application or cartridge from the command line. To start a cartridge or application, access the URL for that cartridge or application using your browser.

Syntax

```
owscctl start [ -T ] [ -w WRB_Process... ] [ -l Listener_Name... ]  
[ -orb ] [ -i No_Of_Instances ] [ -force ] [ -nodemgr ]  
[ -s Site_Name ] [ -n Node_Name ]
```

Arguments

-T	“Talkative” — this argument puts owscctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify -T if you want owscctl to operate in verbose mode.
-w <i>WRB_Process</i>	Starts the specified process. Valid values for <i>WRB_Process</i> are: all — all WRB listener processes oassrv — the OAS server process
-l <i>Listener_Name</i>	Starts the specified listener. You may use the argument all to start all configured listeners at once: owscctl start -l all This command starts all configured listeners on all nodes in your site. Use the -n <i>Node_Name</i> argument to restrict this command to a particular node.
-orb	Starts up the ORB
-i <i>No_of_Instances</i>	Starts multiple instances of a process. This option must be used with the -w <i>WRB_Process</i> option described above. For example: owscctl start -w wrblog -i 2 will start two instances of the wrblog process. You may not start more than one instance of wrbroker , wrbmon , or wrbcfg .

-force	<p>Used only on remote nodes to:</p> <ul style="list-style-type: none"> - stop a single listener on that node - start a single listener on that node - start a single WRB process on that node <p>Most of the time, start and stop commands are issued from the primary node, and are governed by rules set in the site.app file. Since remote nodes do not have a site.app file, you must use the -force option to perform the above tasks on a remote node. For example:</p> <pre>owscctl start -l mylistener -force</pre> <p>will start a listener called mylistener on a remote node.</p>
-nodemgr	<p>Used to manage the Node Manager Listener. The Node Manager Listener allows you to connect to the database and manage nodes, but does not have any WRB connections. The Node Manager Listener must be started first, before you can start the administration listener, or any other listeners.</p>
-s <i>Site_Name</i>	<p>Starts the following components in this order, on the specified site:</p> <ul style="list-style-type: none"> - all ORB processes - all WRB processes - all configured listeners, except the Node Manager Listener
-n <i>Node_Name</i>	<p>When used alone, starts the following processes on the specified node:</p> <ul style="list-style-type: none"> - all ORB processes - all WRB processes - all configured listeners, except the Node Manager Listener <p>When used in conjunction with the -l <i>Listener_Name</i> argument, -n <i>Node_Name</i> restricts the command to the specified node.</p>
no argument	<p>If no argument is specified after the start command, owscctl will start up all ORB processes, all WRB processes, and all configured listeners for the site.</p>

stop

Purpose

Used to stop processes from the command line.

Syntax

```
owscctl stop [ -T ] [ -w WRB_Process... ] [ -l Listener_Name... ]
[ -orb ] [ -c Cartridge_Name... ] [ -a Application_Name... ]
[ -i Instance_No ] [ -p Instance_PID ] [ -force ] [ -nodemgr ]
[ -s Site_Name ] [ -n Node_Name ]
```

Arguments

-T	<p>“Talkative” — this argument puts owscctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify -T if you want owscctl to operate in verbose mode.</p>
-w WRB_Process	<p>Stops the specified process. Valid values for <i>WRB_Process</i> are:</p> <ul style="list-style-type: none"> <i>all</i> — all WRB listener processes <i>oassrv</i> — the OAS server process <p>Note: The WRB must be stopped by the same user who started it.</p>
-l Listener_Name	<p>Stops the specified listener. You may use the argument <i>all</i> to stop all configured listeners at once:</p> <pre>owscctl stop -l all</pre> <p>This command stops all configured listeners on all nodes in your site. Use the <i>-n Node_Name</i> argument to restrict this command to a particular node.</p> <p>Note: Listeners must be stopped by the same user who started them.</p>
-orb	<p>Shuts down the ORB.</p>
-c Cartridge_Name	<p>Stops the specified cartridge. For example:</p> <pre>owscctl stop -c myapp/mycart</pre> <p>will stop the cartridge named mycart which runs in the myapp application. If you stop a cartridge, the application to which it belongs will still be running.</p>

-a <i>Application_Name</i>	<p>Stops the specified application and all cartridges in that application. For example:</p> <pre>owsctl stop -a myapp</pre> <p>will stop the application named <code>myapp</code>, and all associated cartridges in <code>myapp</code>.</p>
-i <i>Instance_No</i>	<p>Stops the specified instance. This option must be used with the <code>-w <i>WRB_Process</i></code> or <code>-l <i>Listener_Name</i></code> option described above. For example:</p> <pre>owsctl stop -w wrblog -i 0</pre> <p>will stop the first instance of the <code>wrblog</code> process.</p>
-p <i>Instance_PID</i>	<p>Stops the specified process. This option must be used with the <code>-w <i>WRB_Process</i></code> or <code>-l <i>Listener_Name</i></code> option described above. For example:</p> <pre>owsctl stop -w wrblog -p 1234</pre> <p>will stop the instance of the <code>wrblog</code> process with process ID 1234.</p>
-force	<p>This argument is used only on remote nodes to:</p> <ul style="list-style-type: none"> - stop a single listener on that node - start a single listener on that node - start a single WRB process on that node <p>Most of the time, start and stop commands are issued from the primary node, and are governed by rules set in the site.app file. Since remote nodes do not have a site.app file, you must use the <code>-force</code> option to perform the above tasks on a remote node. For example:</p> <pre>owsctl stop mylistener -force</pre> <p>will stop a listener called <code>mylistener</code> on a remote node.</p>
-nodemgr	<p>Used to manage the Node Manager Listener. The Node Manager Listener allows you to connect to the database and manage nodes, but does not have any WRB connections. The Node Manager Listener should be shut down last, after you have shut down the administration listener, and any other listeners.</p>
-s <i>Site_Name</i>	<p>Stops the following components on the specified site:</p> <ul style="list-style-type: none"> - all configured listeners, except the Node Manager Listener - all WRB processes - all ORB processes

-n <i>Node_Name</i>	<div>Stops the following processes on the specified node:<ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes<div>When used in conjunction with the -l <i>Listener_Name</i> argument, -n <i>Node_Name</i> restricts the command to the specified node.</div></div>
no argument	<div>If no argument is specified after the stop command, owscctl will start up all ORB processes, all WRB processes, and all configured listeners.</div>

status

Purpose

The status command provides information about the specified process. You use the status command to obtain information about:

- WRB processes
- listeners
- cartridges
- applications
- the Node Manager
- sites
- nodes

The information is displayed in tabular form. The information displayed depends on the type of process. For example, when you query the status of a WRB process, the output is displayed as in the following example:

```
owscctl status -w wrbmon
Name      Description      Node      System ID
wrbmon    "Monitoring Daemon" miaow     381:0
orb       "ORB SErvers"    ----     UP
```

The columns shown are:

- Name — the name of the process
- Description — a brief description of the process

- Node — the hostname of the machine where the process is running
- System ID — the system ID of the process

When you request the status of a listener, you will also see the following information:

- Port/Type — shows the port on which the listener runs, and the type of listener. For example, 8889/Oracle 4.0 indicates an Oracle Web Listener of version 4.0 running on port 8889.

When you request the status of a cartridge or application, you will also see similar information. For a detailed description of status information, refer to the Oracle Application Server Manager online help pages.

Note: Since status is taken at polling intervals, the first time you request the status of a process, it may take as long as eight minutes for the status to become available. Process status is automatically updated every three minutes.

Syntax

```
owscctl status [ -T ] [ -w WRB_Process... ] [ -l Listener_Name... ]
[ -c Cartridge_Name... ] [ -a Application_Name... ]
[ -i Instance_No ] [ -p Instance_PID ] [ -nodemgr ] [ -s Site_Name ]
[ -n Node_Name ]
```

Arguments

-T	“Talkative” — this argument puts owscctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify -T if you want owscctl to operate in verbose mode.
-w <i>WRB_Process</i>	Starts the specified process. Valid values for <i>WRB_Process</i> are: all — all WRB listener processes oassrv — the OAS server process
-l <i>Listener_Name</i>	-l followed by the name of a configured listener gives status information about the specified listener.

-c Cartridge_Name	<p>-c followed by the name of a cartridge gives status information about the specified cartridge. For example:</p> <pre>owscctl status -c myapp/mycart</pre> <p>If you do not provide a specific cartridge name, the status of all cartridges in the specified application will be displayed. For example:</p> <pre>owscctl status -c myapp</pre> <p>will display the status of all cartridges associated with myapp.</p>
-a Application_Name	<p>-a followed by the name of an application gives status information about the specified application. For example:</p> <pre>owscctl status -a myapp</pre>
-p Instance_PID	<p>-p followed by a process ID gives status information about the specified process. This option must be used with the -w <i>WRB_Process</i> or -l <i>Listener_Name</i> option described above. For example:</p> <pre>owscctl status -w wrblog -p 1234</pre> <p>will give status information about the instance of the wrblog process with process ID 1234.</p>
-nodemgr	<p>-nodemgr is used to manage the Node Manager Listener. The Node Manager Listener allows you to connect to the database and manage nodes, but does not have any WRB connections. To obtain status of the Node Manager Listener, type:</p> <pre>>owscctl status -nodemgr</pre> <p>OWS-08806: Oracle Web Listener 'node' is already UP and running at pid 11558.</p>
-s Site_Name	<p>-s followed by a valid site name gives status information about the following components on the specified site:</p> <ul style="list-style-type: none"> - all configured listeners, except the Node Manager Listener - all ORB processes - all WRB processes
-n Node_Name	<p>-n followed by a fully qualified node name gives status information about the following processes on the specified node:</p> <ul style="list-style-type: none"> - all configured listeners, except the Node Manager Listener - all ORB processes - all WRB processes

reload

Purpose

Used to stop and restart processes from the command line.

Syntax

```
owsctl reload [ -T ] [ -w all ] [ -l Listener_Name... ]
[ -c Application_Name... ] [ -a all ] [ -nodemgr ] [ -s Site_Name ]
[ -n Node_Name ]
```

Arguments

<code>-T</code>	“Talkative” — this argument puts owsctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify <code>-T</code> if you want owsctl to operate in verbose mode.
<code>-w all</code>	Stops and restarts all WRB process. Reloading a single WRB process is not supported.
<code>-l <i>Listener_Name</i></code>	Stops and restarts the specified listener. You may use the argument <code>all</code> to reload all configured listeners at once: <pre>owsctl reload -l all</pre> <p>For the Oracle Web Listeners and Apache listeners, the reload command sends a reload signal to the listener, and rereads and updates the configuration file.</p>
<code>-c <i>Application_Name</i></code>	Stops and restarts all cartridges in the specified application. For example: <pre>owsctl reload -c myapp</pre> <p>will stop and restart all cartridges in the <code>myapp</code> application. Reloading individual cartridges is not supported.</p>
<code>-a all</code>	Stops and restarts all applications and all cartridges in those applications. For example: <pre>owsctl reload -a all</pre> <p>will stop and restart all applications in the site, and all associated cartridges.</p>

-nodemgr	Used to start and stop the Node Manager Listener. The Node Manager Listener allows you to connect to the database and manage nodes, but does not have any WRB connections. The Node Manager Listener allows you to use the Oracle Application Server Manager.
-s <i>Site_Name</i>	Stops and restarts the following components on the specified site: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes
-n <i>Node_Name</i>	Stops and restarts the following processes on the specified node: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes
no argument	Stops and restarts the entire site, including: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes

dump

Purpose

Used to display state information about processes. For example:

```
>owscctl dump -w wrbmon -p -1234 -n my-node
      Process Status      Connections
Name  Node   SID    CPU    Memory | Current      Last Time      Total
wrb/wrbmon  miranda 1641:0  0      2904   1      10:03:35      25
```

For a detailed explanation of the meaning of each column, refer to the online help pages for the Oracle Application Server Manager Show State utility.

Syntax

```
owscctl dump [ -T ] [ -w WRB_Process... ] [ -l Listener_Name... ]
[ -c Cartridge_Name... ] [ -a Application_Name... ]
[ -i Instance_No ] [ -p Instance_PID ] [ -s Site_Name ]
[ -n Node_Name ]
```


Arguments

<code>-T</code>	“Talkative” — this argument puts owsctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify <code>-T</code> if you want owsctl to operate in verbose mode.
<code>-w WRB_Process</code>	Starts the specified process. Valid values for <i>WRB_Process</i> are: <code>all</code> — all WRB listener processes <code>oassrv</code> — the OAS server process
<code>-l Listener_Name</code>	Shows state about the specified listener. You may use the argument <code>all</code> to display state for all configured listeners at once: <code>owsctl dump -l all</code>
<code>-c Cartridge_Name</code>	Displays state information about the specified cartridge. For example: <code>owsctl dump -c myapp/mycart</code> will display state information about the cartridge named <code>mycart</code> .
<code>-a Application_Name</code>	Displays state information about the specified application and all cartridges in that application. For example: <code>owsctl dump -c myapp</code> will display state information about the application named <code>myapp</code> , and all associated cartridges in <code>myapp</code> .
<code>-i Instance_No</code>	Displays state information about the specified instance of a WRB process. This option must be used with the <code>-w WRB_Process</code> option described above. For example: <code>owsctl dump -w wrblog -i 0</code> will display state information about the first instance of the <code>wrblog</code> process.
<code>-p Instance_PID</code>	Displays state information about the specified process. This option must be used with the <code>-w WRB_Process</code> or <code>-l Listener_Name</code> option described above. For example: <code>owsctl dump -w wrblog -p 1234</code> will display the state of the instance of the <code>wrblog</code> process with process ID 1234.

<code>-s Site_Name</code>	Displays the state of the following components on the specified site: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes
<code>-n Node_Name</code>	Displays the state of the following processes on the specified node: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes
	This option is required.

monitor

Purpose

For a given process, the monitor command returns monitoring information such as:

- the amount of time used by the process in 1/100th of a second
- the number of current in-bound connections to the process
- the number of pending and completed requests

Note: All monitoring statistics are a rough indication of the state of the server. They are not precise. For example, the total number of requests completed include internal calls of Oracle Application Server.

For a detailed description of the monitoring information, refer to the Oracle Application Server Manager help pages for monitoring processes.

Syntax

```
owsctl monitor [ -T ] [ -w WRB_Process... ] [ -l Listener_Name... ]  
[ -orb ] [ -c Cartridge_Name... ] [ -a Application_Name... ]  
[ -i Instance_No ] [ -p Instance_PID ] [ -s Site_Name ]  
[ -n Node_Name ]
```

Arguments

-T	“Talkative” — this argument puts owscctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify -T if you want owscctl to operate in verbose mode.
-w <i>WRB_Process</i>	Starts the specified process. Valid values for <i>WRB_Process</i> are: all — all WRB listener processes oassrv — the OAS server process
-l <i>Listener_Name</i>	Displays monitoring information about the specified listener. You may use the argument all to display information about all configured listeners at once: owscctl monitor -l all
-orb	Displays monitoring information for the ORB.
-c <i>Cartridge_Name</i>	Displays monitoring information about the specified cartridge. For example: owscctl monitor -c myapp/mycart will display monitoring information about the cartridge named mycart.
-a <i>Application_Name</i>	Displays monitoring information about the specified application and all cartridges in that application. For example: owscctl monitor -c myapp will display monitoring information about the application named myapp, and all associated cartridges in myapp.
-i <i>Instance_No</i>	Displays monitoring information about the specified instance of a WRB process. This option must be used with the -w <i>WRB_Process</i> option described above. For example: owscctl monitor -w wrblog -i 0 will display monitoring information about the first instance of the wrblog process.
-p <i>Instance_PID</i>	Displays monitoring information about the specified process. This option must be used with the -w <i>WRB_Process</i> or -l <i>Listener_Name</i> option described above. For example: owscctl monitor -w wrblog -p 1234 will display the monitoring information about the instance of the wrblog process with process ID 1234.

<code>-s Site_Name</code>	Displays monitoring information about the following components on the specified site: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes
<code>-n Node_Name</code>	Displays monitoring information about the following processes on the specified node: <ul style="list-style-type: none">- all configured listeners, except the Node Manager Listener- all WRB processes- all ORB processes

getIOR

Purpose

The `getIOR` command retrieves the Interoperable Object Reference (IOR) for the specified object. Running this command will print a string to your screen which is unique for each object reference. You may pass this string to an API to determine the actual object reference.

Syntax

```
owscctl getIOR [ -T ] [ -w WRB_Process... ] [ -c Cartridge_Name ]  
[ -a Application_Name ] [ -i Instance_No ] [ -p Instance_PID ]  
[ -s Site_Name ] [ -n Node_Name ]
```

Arguments

<code>-T</code>	“Talkative” — this argument puts owscctl in verbose mode. This gives the user more feedback during the execution of commands. This option must be used with at least one other option. Each time you execute a command, you must specify <code>-T</code> if you want owscctl to operate in verbose mode.
<code>-w WRB_Process</code>	Starts the specified process. Valid values for <code>WRB_Process</code> are: <ul style="list-style-type: none"><code>all</code> — all WRB listener processes<code>oassrv</code> — the OAS server process

-c <i>Cartridge_Name</i>	Displays the IOR of the specified cartridge. For example: <pre>owscctl getIOR -c myapp/mycart</pre> will retrieve the IOR of the cartridge named mycart.
-a <i>Application_Name</i>	Displays the IOR of the specified application and all cartridges in that application. For example: <pre>owscctl getIOR -c myapp</pre> will display the IOR of the application named myapp.
-i <i>Instance_No</i>	Displays the IOR of the specified instance. This option must be used with the -w <i>WRB_Process</i> option described above. For example: <pre>owscctl getIOR -w wrblog -i 0</pre> will print the IOR of the first instance of the wrblog process.
-p <i>Instance_PID</i>	Displays the IOR of the specified process. This option must be used with the -w <i>WRB_Process</i> option described above. For example: <pre>owscctl getIOR -w wrblog -p 1234</pre> will display the IOR of the instance of the wrblog process with process ID 1234.

WRB, ORB and Cartridge Processes

The WRB, ORB, and cartridge processes coordinate the distributed inter-process communications for Oracle Application Server.

The ORB process is oasorb. If running the ORB's Dynamic Monitoring Service, the oasomo process will also exist.

The WRB Oracle Web Logger process, **wrblog**, can be started individually and executed with multiple instances.

The cartridges can run on the same primary machine as the WRB processes, or they can run on a remote machine. Multiple cartridge processes can be run on the same machine.

The following examples show how to run the processes:

Starting the WRB Process

Start the WRB processes on the primary machine by entering:

```
owscctl start -w all
```

Displaying the WRB Status

Show the status of WRB processes on the primary node by entering:

```
owscctl status -w all
```

Starting a Listener

Start the Node Manager Listener by entering:

```
owscctl start -nodemgr
```

Start the Oracle administration listener by entering:

```
owscctl start -l admin
```

To start other listeners, type:

```
owscctl start -l <listener_name>
```

Displaying the Listener Status

Show the status of the Oracle administration listener by entering:

```
owscctl status -l admin
```

Starting the Cartridge on a Remote Machine

Start the cartridge process on the remote machine by entering:

```
owscctl start -c <Cartridge_Name>
```

Starting the ORB Processes

Start the ORB processes on the primary machine by entering:

```
owscctl start -orb
```

Starting all WRB Processes

To start all the WRB processes, type:

```
owscctl start -w all
```

Files

Web Listener registration file:

ORAWEB_ADMIN\ORAWEB_SITE\httpd_machine_name\owl.cfg

Table A-1 WRB process files

Windows NT	Unix
%ORACLE_HOME%\ows\4.0\bin\wrbcfg	\$ORAWEB_HOME/bin/wrbcfg
%ORACLE_HOME%\ows\4.0\bin\wrblog	\$ORAWEB_HOME/bin/wrblog
%ORACLE_HOME%\ows\4.0\bin\wrbasrv	\$ORAWEB_HOME/bin/wrbasrv
%ORACLE_HOME%\ows\4.0\bin\wrbahsrv	\$ORAWEB_HOME/bin/wrbahsrv
%ORACLE_HOME%\ows\4.0\bin\wrbroker	\$ORAWEB_HOME/bin/wrbroker
%ORACLE_HOME%\ows\4.0\bin\wrbmon	\$ORAWEB_HOME/bin/wrbmon
%ORACLE_HOME%\ows\4.0\bin\wrbrmpxy	\$ORAWEB_HOME/bin/wrbrmpxy
%ORACLE_HOME%\ows\4.0\bin\wrksf	\$ORAWEB_HOME/bin/wrksf

The owsstat utility

Use the **owsstat** utility to monitor the status of the Web Listener. The **owsstat** utility is a monitoring daemon that sends a URL to a Web Listener that is identified by machine name and port number. If **owsstat** does not receive a reply from the Web Listener, it executes the event trigger script.

Syntax

```
owsstat -h <machine_name> -p <port_number>
[ -s <SSL_port> | -timeout <timeout> | -poll <polling_period> |
-action <trigger_script> | -uri <uri> | -v ]
```

Note: To display the help text for the **owsstat** utility, run **owsstat** without any parameters at the command prompt.

Arguments

-h	Followed by the machine name where the Web Listener is running.
-p	Followed by the port number that the Web Listener is listening to.
-s	Followed by the SSL port number. Add this command switch when monitoring a URL with SSL turned on. The URL should be “https://” instead of “http://.”

-timeout	Followed by the timeout period in seconds. If owsstat does not receive a reply after the timeout period, the event trigger script is executed. If this argument is not provided, the default timeout is 20 seconds.
-action	Followed by the event trigger script or program. If owsstat does not receive a timeout reply, the script or given program is executed by owsstat . If this argument is not provided, the owstrigger.sh script is executed. This script is found in \$ORAWEB_HOME\bin\owstrigger.sh on Windows NT. On Unix, it is found in \$ORAWEB_HOME/bin/owstrigger.sh .
-poll	Followed by the polling period in seconds. The owsstat sends a URL to the Web Listener during every polling period. If this argument is not provided, the default polling period is 120 seconds.
-uri	Followed by the URI used to poll the Web Listener. The URL should not be protected. By default, the OWSSTAT cartridge application owsstat is used. Make sure that the OWSSTAT cartridge is registered with the Oracle Application Server.
-v	Print out the version.

Environment

To use the proxy server, set the `http_proxy` environment variable to the name of the proxy server that is being used.

The following examples show how to run the **owsstat** monitoring daemon.

Example A-1 Monitor the Web Listener on Port 9999

In this example, the **owsstat** tool is monitoring the Web Listener that is running on machine **miranda.us.oracle.com** and listening to port 233. For example:

```
>owsstat -h miranda.us.oracle.com -p 9999
Oracle Web Status Monitor 4.0.8.0.0
Copyright (c) Oracle Corporation 1998. All rights reserved.
Monitor Information:
Machine: miranda.us.oracle.com
Port: 9999
Timeout: 20 seconds
Polling: 120 seconds
Trigger Action: /private/home/apps/oracle/oas4/ows/4.0/bin/owstrigger.sh
URL: http://miranda.us.oracle.com:9999/owsstat
```


Example A-2 Monitor the Web Listener on Port 268

This example uses **owsstat** to monitor the Web Listener running on machine **miranda.us.oracle.com** and listening to port 268 every 300 seconds (5 minutes). If the Web Listener does not respond after 60 seconds, run the user-defined user trigger script **\private\ows\events\emailsysadmin.sh**. For example:

```
>owsstat -h miranda.us.oracle.com -p 268 -poll 300 -timeout 60
-action /private/ows/events/emailsysadmin.sh &
Oracle Web Status Monitor 4.0.8.0.0
Copyright (c) Oracle Corporation 1998. All rights reserved.
Monitor Information:
Machine: miranda.us.oracle.com
Port: 268
Timeout: 60 seconds
Polling: 300 seconds
Trigger Action: /private/ows/events/emailsysadmin.sh &
URL: http://miranda.us.oracle.com:268/owsstat
```

Note: These examples are for Windows NT, but the commands are identical on Unix. Unix uses forward slashes; while Windows NT uses backward slashes when specifying directory paths.

Trigger Environment Variables

On Unix, several environment variables are available with the event trigger script. For example:

Trigger Environment Variables	Types of Errors
OWSSTAT_TRIG_CODE	“No data” — no data received from the server “Connect error” — server is down “Timeout” — no response from server after timeout “Redirection error” — error occurs during redirection “OWSSTAT exit” — owsstat was interrupted and exited “Other HTTP error other than ”200 OK”
OWSSTAT_TRIG_SCPT	Trigger script path name
OWSSTAT_TRIG_TIMEOUT	Timeout period
OWSSTAT_TRIG_POLL	Polling period
OWSSTAT_TRIG_URL	URL being monitored

Files

The default event trigger script is:

ORAWEB_HOME\bin\owstrigger.sh

The **owsstat** process file is:

ORAWEB_HOME\bin\owsstat

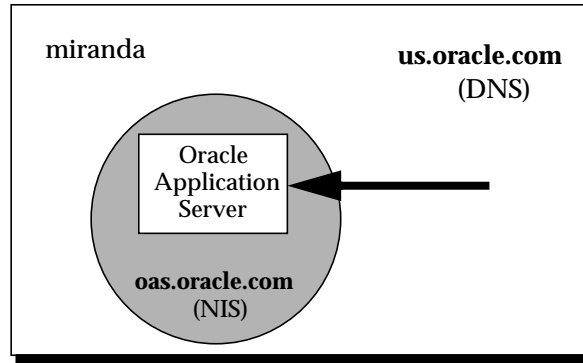
Note: These files are only available on the Unix version of Oracle Application Server Release 4.0.8.

The oasnetconf utility

During installation, Oracle Application Server gathers information about hostnames and domain information directly from the machines. However, you may want to configure these network parameters in Oracle Application Server differently and/or change ports and IP addresses. The **oasnetconf** utility allows you to change the following properties of the Oracle Application Server configuration:

- Hostname
- Domain name

Figure A-1 The node **miranda** belongs to two different domains. In the NIS world it belongs to **oas.oracle.com**, while in the DNS world it belongs to **us.oracle.com**.



Consider a typical scenario (Figure A-1) where the node **miranda** belongs to two different domains — **oas.oracle.com** (NIS) and **us.oracle.com** (DNS). When Oracle Application Server is installed on the node, it reads in values from the NIS domain and configures all administration files. The system functions as long as requests come from the NIS world. However, requests from the DNS world cannot be serviced as all the system configuration information is different. If the system gets requests only from the DNS world then there has to be some means of changing the system administration information to be configured with values from the DNS world. The **oasnetconf** utility provides this capability.

Note: It is assumed that all the hosts in the system have unique hostnames (i.e. we can distinguish between hosts solely by their hostnames without taking their domain names into consideration).

Syntax

```
oasnetconf [ -l | -r <pathname> | -s <sitename> |
-h <OldHostname> | -n <NewHostname> | -d <NewDomain> |
-H <newPrimaryHostName> | -D <newPrimaryDomainName> ]
```

Arguments

<code>-l</code>	Outputs a listing of the current information
<code>-r <pathname></code>	<p>(Optional) Specifies the pathname to the resources.ora file. If the oasnetconf utility is unable to find your resources.ora file, you may need to specify its location.</p> <p>On Windows NT, resources.ora can be found in %ORACLE_HOME%\orb\admin.</p> <p>On Unix, if you have an OFA installation, resources.ora is in \$ORACLE_HOME/orb/admin/<i>sitename</i>.</p> <p>If your Unix installation is non-OFA, resources.ora can be found in \$ORACLE_HOME/orb/4.0/admin.</p>
<code>-s <sitename></code>	(Optional) Specifies the site name. Otherwise, site information is taken from the \$ORAWEB_SITE env variable.
<code>-h <OldHostname></code>	Specifies the old name of the host
<code>-n <NewHostname></code>	Specifies the new name of the host (replaces the old name)
<code>-d <NewDomain></code>	Specifies the new domain of the host (replaces the old domain)
<code>-H <newPrimaryHostName></code>	Specifies the new hostname of the primary node
<code>-D <newPrimaryDomainName></code>	Specifies the new domain name of the primary node

Example A–3 *Changing the hostname and domain name of a remote node*

On the remote node:

```
oasnetconf -s website40 -h oldfoo -n newfoo -d newbar
```

On the primary node:

```
oasnetconf -s website40 -h oldfoo -n newfoo -d newbar
```

Example A–4 *Changing the hostname and domain name of the primary node*

On the primary node:

```
oasnetconf -s website40 -h oldfoo -n newfoo -d newbar
```

or

```
oasnetconf -s website40 -H newfoo -D newbar
```

On ALL remote nodes:

```
oasnetconf -s website40 -H newfoo -D newbar
```

Example A-5 *Changing only the domain of a certain remote node: (changing only the hostname is similar)*

On the remote node:

```
oasnetconf -s website40 -h oldfoo -d newbar
```

On the primary node:

```
oasnetconf -s website40 -h oldfoo -d newbar
```

Example A-6 *Changing only the hostname of the primary node: (changing only the domain is similar)*

On the primary node:

```
oasnetconf -s website40 -h oldfoo -n newfoo
```

or

```
oasnetconf -s website40 -H newfoo
```

On ALL remote nodes:

```
oasnetconf -s website40 -H newfoo newbar
```

The oaspasswd utility

Use the **oaspasswd** utility to change passwords associated with the following:

- Node Manager
- ORB
- Wallet Manager

The passwords for each of the above entities are configurable on every node in the system.

Note: If you want to see the changes, you need to stop the listener and restart it.

This utility is available for UNIX and NT.

Syntax

```
oaspasswd [-h <HostName | ALL ] [-s <site> ]
```

Arguments

-h <HostName>	Changes passwords corresponding to hostname. This can only be specified on the primary node as it requires access to site.app .
- h ALL	Changes passwords of all the hosts in the system. This can only be specified on the primary node.
-s <site>	Specifies the site name. If the site name is not mentioned, it is assumed to be set in the environment (and/or registry for NT).
no argument	Changes all three passwords on the local host.

Restricting Access to the oaspasswd Utility

You should use your operating system to restrict the use of the **oaspasswd** utility.

On Unix,

restrict execute privileges of **\$ORAWEB_HOME/bin/oaspasswd.exe** to only those users who should change passwords.

On Windows NT,

If you use an NTFS file system, follow these steps to restrict permissions for **oaspasswd.exe**.

1. From Windows NT Explorer, navigate to %ORACLE_HOME%\ows\4.0\bin.
2. Select the **oaspasswd** executable.
3. From the File menu, select Properties.
4. Click on the Security tab.
5. Set permissions for the groups which should have execute privileges.

Note: If you use a FAT file system, Windows NT does not support operating system protection of files.

Examples

The following examples show how to use the **oaspasswd** utility.

Example A-7 Setting all the node managers and wallet managers in the system with the same password.

On the primary node (assuming \$ORAWEB_SITE is set in the environment):

```
> oaspasswd -h ALL
NewPassword      :
Confirm Password :
Success: Updated site.app.
Success: Updated listener configuration file.
```

On each remote node (assuming \$ORAWEB_SITE is set in the environment):

```
> oaspasswd
NewPassword      :
Confirm Password :
Success: Updated site.app.
Success: Updated listener configuration file.
```

Example A-8 Setting the node and wallet manager password for a particular remote node

On the primary node:

```
> oaspasswd -s website40 -h remoteNodeName
NewPassword      :
Confirm Password :
Success: Updated site.app.
Success: Updated listener configuration file.
```

On the remote node:

```
> oaspasswd -s website40
NewPassword      :
Confirm Password :
Success: Updated site.app.
Success: Updated listener configuration file.
```

Example A–9 *Setting the node and wallet manager password for the primary node*

On the primary node:

```
> oaspasswd -s website40
NewPassword      :
Confirm Password :
Success: Updated site.app.
Success: Updated listener configuration file.
```

The oasmcastcfg utility

This utility is used to display or change the Group Messaging Configuration that was set during installation.

For more information on Group Messaging Configuration see the *Oracle Application Server Installation Guide*.

Syntax

```
oasmcastcfg show

or

oasmcastcfg set [-i IP_address] [-p port]
```

Arguments

-i <i>IP_address</i>	The IP address of the messaging host. The address must be between 225.0.0.0 and 239.255.255.255.
-p <i>port</i>	The port number on the messaging host that is used for messaging. The port number must be between 1024 and 6553. It is recommended that you use a value greater than 49151 since these ports are not typically used by other applications.

Examples

Example A–10 *Displaying the current configuration*

This will display the current group messaging settings.

```
% oasmcastcfg show
Multicast IP Address      235.0.0.0
```



```
Multicast Port          50500
The command completed successfully on host myhost.
```

Example A-11 Changing only the IP address

This will only change the IP address. The previous port will still be used.

```
% oasmcastcfg set -i 235.0.0.10
Please wait while the command is being processed on host myhost ...
The command completed successfully on host myhost.
```

Example A-12 Changing only the port number

This will only change the port number. The previous IP address will still be used.

```
% oasmcastcfg set -p 50501
Please wait while the command is being processed on host myhost ...
The command completed successfully on host myhost.
```

Example A-13 Changing both the IP address and port

This will change both the IP address and port number used for group messaging. Since both the IP address and port are set in one command, it is treated as two requests. This causes the “Please wait...” message to appear twice.

```
% oasmcastcfg set -i 235.0.0.10 -p 50501
Please wait while the command is being processed on host myhost ...
The command completed successfully on host myhost.
Please wait while the command is being processed on host myhost ...
The command completed successfully on host myhost.
```

Configuration and Log Files

Contents

- [Windows NT Files](#)
- [Unix Files](#)

Windows NT Files

The following tables describe the various configuration and log files for an NT installation:

Table B–1 *OAS installer log messages (Windows NT)*

File	Default Location	Description
install.log	%ORACLE_HOME%\orainst\install.log	log of installation process
make.log	%ORACLE_HOME%\orainst\make.log	log of relink information (saved when products are installed, added or deleted)
sql.log	%ORACLE_HOME%\orainst\sql.log	log of all SQL statements executed during the installation

Table B–2 OAS log messages (Windows NT)

File	Default Location	Description
<code>svlistenerName.err</code>	<code>%ORACLE_HOME%\ows\admin\website\httpd_machineName\listenerName\svlistenerName.err</code>	Spyglass listener error messages
<code>svlistenerName.log</code>	<code>%ORACLE_HOME%\ows\admin\website\httpd_machineName\listenerName\svlistenerName.log</code>	Spyglass listener log messages
<code>wrb.log</code>	<code>%ORACLE_HOME%\ows\admin\website\log\wrb.log</code>	logs all OAS processes, including cartridges
<code>xfp.log</code>	<code>%ORACLE_HOME%\ows\admin\website\log\xfp.log</code>	log of HTTP clients accessing listeners

Table B–3 OAS configuration files (Windows NT)

File	Default Location	Description
<code>owl.cfg</code>	<code>%ORACLE_HOME%\ows\admin\website\httpd_machineName\owl.cfg</code>	contains a list of registered OAS listeners and their respective configuration settings
<code>site.app</code>	<code>%ORACLE_HOME%\ows\admin\website\wrb\site.app</code>	OAS site specific configuration file
<code>svlistenerName.cfg</code>	<code>%ORACLE_HOME%\ows\admin\website\httpd_machineName\listenerName\svlistenerName.cfg</code>	listener configuration file
<code>wrb.app</code>	<code>%ORACLE_HOME%\ows\admin\website\wrb\wrb.app</code>	OAS process and cartridge configuration file

Table B–4 ORB related files (Windows NT)

File	Default Location	Description
<code>acl_files</code>	<code>%ORACLE_HOME%\orb\admin\acl_file</code>	specifies protection for CORBA IDL methods
<code>logs\</code>	<code>%ORACLE_HOME%\orb\admin\logs\</code>	directory containing ORB log files
<code>names\</code>	<code>%ORACLE_HOME%\orb\admin\names\</code>	directory containing CORBA Name Server specific data files
<code>resources.ora</code>	<code>%ORACLE_HOME%\orb\admin\resources.ora</code>	configuration file for ORB

Unix Files

The following tables describe the various configuration and log files for a Unix installation:

Table B–5 OAS installer log messages (Unix)

File	Default Location	Description
install.log	\$ORACLE_HOME/orainst/install.log	log of installation process
make.log	\$ORACLE_HOME/orainst/make.log	log of relink information (saved when products are installed, added or deleted)
os.log	\$ORACLE_HOME/orainst/os.log	log of operating system responses to Unix commands
sql.log	\$ORACLE_HOME/orainst/sql.log	log of all SQL statements executed during the installation

Table B–6 OAS log messages (Unix)

File	Default Location	Description
svlistenerName.err	\$ORAWEB_ADMIN/website/httpd_machineName/listenerName/svlistenerName.err	Spyglass listener error messages
svlistenerName.log	\$ORAWEB_ADMIN/website/httpd_machineName/listenerName/svlistenerName.log	Spyglass listener log messages
wrb.log	(Non-OFA) \$ORACLE_HOME/ows/admin/website/log/wrb.log (OFA) \$ORACLE_BASE/admin/ows/website/log	logs all OAS processes, including cartridges
xlfi.log	\$ORACLE_HOME/ows/4.0/admin/website/log/xlfi.log	log of HTTP clients accessing listeners

Table B–7 OAS configuration files (Unix)

File	Default Location	Description
owl.cfg	\$ORAWEB_ADMIN/website/httpd_machineName/owl.cfg	contains a list of registered OAS listeners and their respective configuration settings
site.app	\$ORAWEB_ADMIN/website/wrb/site.app	OAS site specific configuration file
svlistenerName.cfg	\$ORAWEB_ADMIN/website/httpd_machineName/listenerName/svlistenerName.cfg	listener configuration file
wrb.app	\$ORAWEB_ADMIN/website/wrb/wrb.app	listener configuration file

Table B–8 *ORB related files (Unix)*

File	Default Location	Description
acl_files	\$ORACLE_HOME/orb/4.0/admin/acl_file	specifies protection for CORBA IDL methods
logs/	\$ORACLE_HOME/orb/4.0/admin/logs/	directory containing ORB log files
names/	\$ORACLE_HOME/orb/4.0/admin/names/	directory containing CORBA Name Server specific data files
resources.ora	\$ORACLE_HOME/orb/4.0/admin/resources.ora	configuration file for ORB

Index

A

accessing

- database on remote host, 6-2
- Oracle Application Server Manager, 2-5
- Welcome page, 1-5

adding

- Add CWeb Cartridge, 9-7
- Add JWeb Cartridge, 9-8, 9-9
- Add Listener form, 3-6
- Add LiveHTML Cartridge, 9-10
- Add Node form, 2-8
- Add operation, 1-7
- Add Perl Cartridge, 9-12
- cartridges, 9-4
- ECO/Java application form, 8-10

administering

- HTTP listener, 3-3
- Oracle Application Server, 2-9
- Web Request Broker (WRB), 2-9

Administration Utility listener

- overview, 2-4
- starting, 2-5
- status, 2-4

Apache

- listener, 3-25
- migration, 4-1
- performance, 4-14
- using with JCO cartridge, 4-14

application, 1-1

- add application form, 8-8, 8-9
- Add C++ CORBA application, 8-12
- adding, 8-7, 8-13, 8-14
- remote node, 8-14

Application Instance Security form, 9-32

Application Instance Tuning form, 9-31

Application Instances Parameters form, 9-34

application monitoring form, 8-30

cartridge-based, 7-2

configuration, 7-2, 7-4, 8-1 to 8-3, 8-7

object parameters, 8-3

roadmap, 8-6

ECO/Java, 8-3

EJB, 8-3

instance, 7-4, 8-3

introduction, 7-1

managing, 8-27

MIME types, 8-23

parameters, 8-2, 8-4

priority, 8-19

programming models, 7-6

request-response model, 7-6

session model, 7-6

transaction model, 7-7

remote node, 8-13, 8-14

applications

C++ CORBA, 7-5

architecture, 1-1

programming models, 7-6

archiving

logging, 10-3

max file size, 10-5

rollover, 10-3

attribute logging, 10-9

authentication, 3-23, 4-9

B

batch logging, 10-3
browsing a database
 See database browser

C

C++ CORBA
 Add C++ CORBA Application, 8-12
C++ CORBA cartridge, 7-5
CA Roots form, 3-22
cartridge
 C++ CORBA, 7-5
cartridges, 1-1, 9-12, 9-36
 Add JWeb Cartridge, 9-8, 9-9
 Add LiveHTML Cartridge, 9-10
 Add Perl Cartridge, 9-12
 adding, 9-4
 from file, 9-15
 manually, 9-7, 9-8, 9-10, 9-13
 adding manually, 9-8, 9-9
 Cartridge min/max Tuning form, 9-18
 Cartridge Parameters form, 9-24
 Cartridge priority Tuning form, 9-17
 Cartridges Add Application form, 8-10
 Cartridges Monitoring form, 9-38
 changing display name, 9-40
 configuration, 7-2, 7-4, 8-2
 configuration roadmap, 9-3
 configuring, 9-16
 definition, 9-2
 displaying cartridge information, 9-40
 form, 9-36
 instance, 7-2
 introduction, 9-2
 IP/domain, 9-22
 LiveHTML parameters, 9-24
 managing and configuring, 9-1
 monitoring, 9-37
 parameters, 8-3
 Perl parameters, 9-27
 PL/SQL parameters, 9-28
 priority, 8-19
 protections, 9-20

 realm, 9-21
 scheme, 9-21
 server, 7-2, 8-19
 form, 8-18, 8-19
 shared object, 9-7
 stopping, 8-34
certificate
 authority roots, 3-22
 CA Roots form, 3-22
 certificate, 3-22
 client certificate parameter, 8-23
 client sessions parameter, 8-24
CLASSPATH variable, 8-26
CLF logging, 10-29
client sessions parameter, 8-24
command line utilities, A-1
 oasnetconf, A-22
 oaspasswd, A-25
 owsctl, A-2
 owsstat, A-19
commit, 11-2, 11-5
components
 components, 1-1
 configuration roadmap, 9-3
 revealing, 1-6
configuring, 8-28, 9-22, 9-24
 application, 8-1, 8-3, 8-6
 basic steps, 1-3
 cartridge server form, 8-18
 cartridges, 9-1, 9-16
 changing, 8-15
 checklist, 9-3
 client-defined attributes, 10-11, 10-17
 client-defined attributes logging, 10-29
 DADs, 6-1
 DTC, 11-14
 ECO/Java, 8-3
 EJB, 8-3
 encoding extensions, 3-21
 language extensions, 3-19
 listener parameters, 3-7
 load balancing, 8-19
 MIME types, 3-20
 network parameters, 3-8
 object parameters, 8-3

- ORB components, 5-2
- OTS, 11-6, 11-7
- priority, 8-19
- RM, 11-8
- server parameters, 3-13
- user/group parameters, 3-15
- virtual paths, 9-19
- connect string, 6-1
- CORBA, 5-2, 9-2
- CORBA/IIOP requests, 7-8
- CWeb cartridge
 - cartridge entry point, 9-7
 - cartridge shared object, 9-7
 - Parameters form, 9-24

D

DAD

- adding, 6-3
- adding advanced features, 6-11
- advanced form, 6-12
- configure to be a RM, 11-9
- deleting, 6-3
- form, 6-2
- heterogeneous OTS agent, 6-10
- modifying, 6-11
- NLS, 6-11
- operating system authentication, 6-5
- ops\$ user ids, 6-5
- overview, 6-1
- specifying table space, 6-11
- transactional, 6-6, 6-8
 - configuration, 11-9
- transactions form, 6-8
- troubleshooting, 6-7
- database
 - accessing on a remote host, 6-2
 - connecting to, 6-1
 - files, 10-54
 - free space, 10-55
 - logging, 10-1
 - operating system authentication, 6-5
 - ops\$ user ids, 6-5
 - sessions, 10-53
 - space usage, 10-55

- transactional, 11-8
- users, 10-52
- database browser, 10-48
 - files, 10-54
 - form, 1-10, 10-49
 - installing, 10-26
 - rollback segments, 10-53
 - sessions, 10-53
 - tables and views, 10-50
 - tablespace, 10-55
 - free space, 10-55
 - users, 10-52
 - viewing PL/SQL, 10-51
 - web browser support, 10-50
- declarative transactions, 11-22
- default MIME type, 3-14
- Delete operation, 1-7
- Directory form, 3-17
- directory mappings
 - configuring, 3-17
- display name, 8-10
- Distributed Transaction Coordinator
 - See DTC
- Domain Name field, 2-8
- DTC, 11-6
 - configuration, 11-14
 - deleting, 11-30
 - displaying configuration, 11-30
 - startup, 11-28

E

- ECO/Java
 - add application form, 8-10
 - adding
 - ECO/Java application form, 8-11
 - parameters, 8-3, 8-4
- EJB
 - Add EJB Application form, 8-10
 - applications, 7-9
 - parameters, 8-3, 8-4
 - server, 7-9
- Enable ICX Tag parameter, 9-26
- Enable LiveHTML parameter, 9-25
- encoding extensions

- configuring, 3-21
- Encoding form, 3-21
- environment variables
 - CLASSPATH, 8-26
 - environment variables form, 8-25
 - JAVA_HOME, 8-27
 - LD_LIBRARY_PATH, 8-26
 - PATH, 8-26
 - THREADS_FLAG, 8-27
- error page parameter, 8-25
- external listener
 - form, 1-9
 - overview, 1-9

F

- failure
 - detection, 12-3
 - recovery
 - architecture, 12-2
 - components, 12-1, 12-4
- fields
 - Password, 2-9
 - Port, 2-8
 - User, 2-9
- files
 - database, 10-54
- forms, 1-9
 - accessing, 1-6
 - DAD, 6-3, 6-8, 6-12
 - add application form, 8-8, 8-9, 8-10, 8-13
 - display name, 8-10
 - Add Cartridge form, 9-5
 - Add Cartridge form from file, 9-15
 - Add CWeb cartridge form, 9-7
 - Add ECO/Java application form, 8-10, 8-11
 - Add EJB Application form, 8-10
 - Add JWEB cartridge form, 9-8, 9-9
 - Add Listener form, 3-6
 - Add LiveHTML cartridge form, 9-10
 - Add Node form, 2-8
 - Add Perl cartridge form, 9-12
 - Add PL/SQL cartridge form, 9-13
 - Application Instance Security form, 9-32
 - Application Instance Tuning form, 9-31

- Application Instances Parameters form, 9-34
- application monitoring form, 8-30
- Application priority tuning form, 8-19
- Applications Monitoring form, 8-30
- Attribute, 10-11, 10-17
- CA Roots form, 3-22
- Cartridge Add Application form, 8-10
- Cartridge Information form, 9-40
- Cartridge min/max Tuning form, 9-18
- Cartridge Parameters form, 9-24
- Cartridge priority Tuning form, 9-17
- cartridge server form, 8-18, 8-19
- Cartridge Server Hosts form, 8-20
- Cartridges form, 9-35
- Cartridges Monitoring form, 9-38
- configuration form, 8-28
- creating a DAD, 6-2
- CWeb Parameters form, 9-24
- database browser form, 1-10
- Directory form, 3-17
- Display Name form, 9-40
- Encoding, 3-21
- environment variables form, 8-25
- External Listener, 1-9
- hosts configuration form, 8-19
- hosts form, 8-19, 8-20
- HTTP Listeners, 3-3
- HTTP Listeners form, 3-3
- Install, 1-9
- java environment form, 8-27
- Language form, 3-19
- LiveHTML cartridge configuration form, 9-25
- Log Analyzer, 1-9
- logging, 10-2
- logging form, 8-21
- MIME, 3-20
- Network form, 3-8
- OAS Utility, 1-9
- OAS Utility Upgrade, 1-9
- Oracle Application Server form, 2-2, 2-10
- Oracle Application Server monitoring
 - form, 2-11
- Perl Cartridge Configuration form, 9-27
- PL/SQL parameters form, 9-29
- Security form, 3-23

- Server, 3-13
- server form, 8-18
- Site Manager, 1-8
- SSL Certificate Authentication Roots form, 3-22
- web configuration form, 8-23
- XLF Log Entries, 10-31

G

- getting started, 1-3
- granting roles to DAD user, 6-5, 10-28
- group messaging
 - oasmcastcfg utility, A-28

H

- Help operation, 1-7
- Heterogeneous OTS Agent
 - See HO Agent
- HO Agent, 11-6, 11-18
- hosts
 - configuration form, 8-19
 - form, 8-19
 - parameter, 8-20
- HS keyword, 11-18
- HTTP listener, 3-10
 - administration, 3-3
 - connections on multiple ports, 3-10
 - creating, 3-5
 - form, 3-3
 - managing, 3-1
 - monitoring, 3-4
 - parameters, 3-8
- HTTP requests, 7-8

I

- icons, 1-7
- ICX
 - Enable ICX Tag parameter, 9-26
- IIOP, 5-2
- IIS migration, 4-1
- initial file, 3-13
- initial start-up instances parameter, 8-20
- Installation Type field, 2-9

- installing
 - database browser, 10-26
 - Log Analyzer, 10-25
 - PL/SQL toolkit, 10-23
 - utilities, 10-21
- introduction, 1-1
- IP/domain, 9-22

J

- java environment form, 8-27
- JAVA_HOME variable, 8-27
- JCO cartridge
 - using with Apache, 4-14

L

- language extensions
 - configuring, 3-19
- Language form, 3-19
- LD_LIBRARY_PATH variable, 8-26
- LISTENER.ORA file, 11-18
- listeners, 1-1
 - adding, 3-5
 - Administration Utility listener, 2-4
 - checking status, 2-4
 - configuring parameters, 3-7
 - default file, 3-13
 - external listener form, 1-1
 - HTTP, 3-3
 - manager, 2-3
 - managing HTTP, 3-1
 - MIME type, 3-14
 - Node Manager, 1-6, 2-3
 - parameters, 3-8
 - ports, 3-10
 - ports and addresses, 3-1
 - starting, 8-34
 - stopping, 8-34
 - third party, 3-25
 - user directory, 3-13
 - user/group parameters, 3-15
- LiveHTML
 - cartridge configuration form, 9-25
 - Parameters form, 9-25

- load balancing, 1-9
 - configuration, 8-19
- Log Analyzer, 10-28
 - concurrent user report, 10-45
 - displaying log entries, 10-31
 - installing, 10-25
 - log analyzer form, 1-9
 - overview, 1-9
 - system messages report, 10-42
- logging, 10-1
 - archive intervals, 10-3
 - attribute, 10-9
 - batch, browsing a database
 - See database browser, 10-3
 - directory parameter, 8-22
 - displaying stored entries, 10-31
 - file parameter, 8-22
 - form, 8-21
 - mask numbers, 10-6
 - max log file size, 10-5
 - modifying concurrent user report, 10-45
 - parameter, 8-22
 - severity levels, 10-8
 - system messages, 10-42
 - tables, 10-22
 - installing, 10-29
 - troubleshooting, 10-56
 - types, 10-29
 - utilities, 10-1
 - installation form, 1-9
 - XLF, 10-14

M

- manager listener, 2-3
- managing
 - OAS Manager navigational tree, 2-6
 - Oracle Application Server, 2-9, 9-1
 - processes, 2-2
 - web site, 2-1
- mask numbers, 10-6
- max servers parameter, 8-20
- max session idle time parameter, 8-24
- messaging
 - oasmcastcfg utility, A-28

- Microsoft Internet Information Server, 3-25
- migrate form, 1-9
- migrating third-party servers, 4-1
- MIME
 - form, 3-20
 - types
 - configuring, 3-20
 - types parameter, 8-23
- min servers parameter, 8-20
- min/max based load balancing, 1-9
- models, application programming, 7-6
- Modes
 - Stateful and Stateless Modes, 9-43
- monitor command, A-14
- monitoring, 2-11, A-14
 - application monitoring form, 8-30
 - cartridges, 9-37
 - components, 10-11, 10-17
 - HTTP listeners, 3-4
 - Monitor operation, 1-7
 - owsctl command, A-14
- multiports
 - configuring, 3-11
 - definition, 3-1
 - managing, 3-2

N

- navigation tree, 1-6
 - expanding, 1-6
 - OAS Manager, 2-6
 - website level, 1-6
- Net8 service name, 6-1
- Netscape
 - Enterprise Server, 3-25, 4-1
 - FastTrack listener, 3-25
 - registering, 4-11
- Network form, 3-8
- Node Manager
 - icons, 1-7
 - listener
 - entering, 1-6
 - overview, 2-3
 - port number, 2-8
 - starting and stopping, 2-3

- status, 2-3
- Node Name field, 2-8
- nodes, 2-6
 - Add Node form, 2-8
 - transactions on remote nodes, 11-6
 - types, 2-1

O

OAS

See also Oracle Application Server

OAS Manager navigational tree, 2-6

OAS utilities

- external listener, 1-9
- Log Analyzer, 1-9
- overview, 1-8, 1-9
- upgrading, 1-9

oasmcastcfg utility, A-28

oasnetconf utility, A-24

- arguments, A-24
- overview, A-22
- syntax, A-23

oaspasswd utility, A-25

- syntax, A-26

object parameters

- configuration, 8-3

Object Request Broker. *See* ORB

one phase commit, 11-5

operating system authentication, 6-5

ops\$ user ids, 6-5

option

- Add Application Form Option, 8-9

Oracle Application Server

- administration, 2-9
- architecture, 1-1
- command line utilities, A-1
- components, 1-1
- form, 2-2, 2-10
- functions, 2-5
- getting started, 1-3
- Manager, 1-5, 1-6
- Manager accessing, 2-5
- managing, 2-9, 9-1
- monitoring, 2-11
- Node Manager, 1-5, 1-6

- overview, 1-1

processes

- starting, 2-11
- processes configuring, 2-9
- transactions, 11-1
- understanding, 2-1
- Utilities, 1-5, 1-8
- Welcome page, 1-4

ORACLE_SID, 6-5

ORB

- configuration, 5-2
- daemon
 - port number, 5-3
- defined, 5-1
- management, 5-4
- starting, 5-4

OTS

- component overview, 11-4
- configuration, 11-7
- configure, 11-6
- managing environment, 11-28
- Server, 11-7
- startup, 11-28

overview, 1-1

- external listener, 1-9
- managing and configuring cartridges, 9-1
- OAS Manager listeners, 2-3
- transactions, 11-2

owsctl utility

- cartridges, A-17
- command
 - dump, A-12
 - monitor, A-14
 - reload, A-11
 - start, A-4
 - status, A-8
 - stop, A-6
 - version, A-3

- files, A-19

- overview, A-2

- syntax, A-3

owsstat utility

- files, A-22
- overview, A-19
- syntax, A-19

P

parameters

- application, 8-2
 - ECO/Java, 8-4
 - EJB, 8-4
 - application instance, 8-4
 - cartridge, 8-3
 - cartridge timeout, 8-18
 - client certificate, 8-23
 - client sessions, 8-24
 - configuring cartridges, 9-19
 - configuring network, 3-8
 - configuring server, 3-13
 - configuring user/group, 3-15
 - CWeb Parameters form, 9-24
 - default file name, 3-13
 - ECO/Java, 8-3
 - EJB, 8-3
 - error page, 8-25
 - hosts, 8-20
 - initial start-up instances, 8-20
 - logging, 8-22
 - logging directory, 8-22
 - logging file, 8-22
 - max servers, 8-20
 - max session idle time, 8-24
 - MIME types, 8-23
 - min servers, 8-20
 - Perl Parameters form, 9-27
 - priority, 8-19
 - severity level, 8-22
 - tuning, 9-17
 - user directory, 3-13
- ### passwords
- changing, A-25
- ### path
- for Perl libraries, 9-27
- ### PATH variable, 8-26
- ### performance
- Apache, 4-14
- ### Perl
- Add Perl Cartridge, 9-12
 - cartridge parameters, 9-27
 - library path, 9-27

permissions

- configuring, 3-15

PL/SQL

- Add PL/SQ Cartridge, 9-13
- Configuration parameters for a PL/SQL
 - application, 8-2
- Parameters form, 9-29
- toolkit
 - installing, 10-23

ports and addresses

- default settings, 1-2
- duplicate, 3-2
- multiple, 3-1

priority based load balancing, 1-9

processes

- configuring, 2-9
- failure
 - See* failure recovery
- management, 2-2
- recovery
 - authentication servers, 12-6
 - broker/RM, 12-5
 - cartridge server factories, 12-5
 - cartridge servers, 12-6
 - configuration providers, 12-6
 - dispatchers/listeners, 12-5
 - loggers, 12-6
 - monitoring daemon, 12-4
 - RM Proxy, 12-5
- starting, 2-11
- protection, 9-20

R

realm, 9-21

recovery

- See* failure recovery

refreshing, 1-8

registering third-party listeners, 4-1

reloading, 1-8

remote node

- adding, 2-7

remote nodes, 11-6

- adding applications, 8-14
- applications, 8-13, 8-14

- transactions, 11-6
- reports
 - concurrent user, 10-45
 - system messages, 10-42
- request-response model, 7-6
- requests
 - path, 1-2
- restriction schemes, 3-23
- RM, 11-6
 - configuration, 11-8
 - proxy, 7-8, 7-9
 - relationship to DTC, 11-5
 - Resource Manager, 11-8
 - resource manager, 6-10
 - Resource Manager. See RM
 - shutdown, 11-28
 - startup, 11-28
- roadmap, 1-3
- rollback
 - segments, 10-53
 - terminating transaction, 11-2
- root certificates, 3-22
- runtime processes, 1-1

S

- sample path of a request, 1-2
- scheme, 9-21
- secondary node
 - adding applications, 8-13, 8-14
- security, 3-22
 - Security form, 3-23
- server
 - parameters, 3-13
 - Server form, 3-13
- sessions
 - database, 10-53
 - idle time, 8-24
 - model, 7-6
- severity level
 - logging, 10-8
 - parameter, 8-22
- show state (dump), A-12
- Site Manager
 - form, 2-1

- using, 1-8
- SQL*Net v2 connect string, 6-1
- SSL, 3-22
 - SSL Certificate Authentication Roots form, 3-22
- Start operation, 1-7
- Stateful and Stateless Modes, 9-43
- statistics
 - state of server, 3-4
- Stop operation, 1-7
- system messages
 - logging, 10-4, 10-29
 - severity level, 10-8

T

- tablespace
 - free space, 10-55
 - space usage, 10-55
- TCP/IP port
 - connections, 3-10
 - Node Manager listener, 2-8
- third-party listeners, 3-25, 4-1
- THREADS_FLAG variable, 8-27
- TNS_ADMIN, 6-3
- TNSNAMES.ORA file, 6-2, 11-18
- transactions
 - application, 11-5
 - DAD, 6-6
 - DADs, 6-8
 - database, 11-8
 - declarative, 11-22
 - definition, 11-1, 11-2
 - model, 7-7
 - overview, 11-2
 - remote nodes, 11-6
 - shutdown application, 11-28
 - termination, 11-2
- troubleshooting
 - DADs, 6-7
 - logging, 10-56
- Tuning form
 - Min/Max based load balancing, 9-18
 - Priority based load balancing, 9-17
 - tuning form, 1-9
- two phase (2pc) commit, 11-5

U

- Update Page operation, 1-7
- upgrading
 - from previous versions, 1-9
 - overview, 1-9
- uploading log files, 10-29
- User, 3-16
- User/Group form, 3-15
- users
 - database, 10-52
 - operating system authentication, 6-5
 - ops\$ user ids, 6-5
- utilities, 1-8, 10-21
 - oasmcast, A-28
 - oasnetconf, A-22
 - oaspasswd, A-25
 - owsctl, A-2
 - owsstat, A-19

V

- viewing
 - database users, 10-50
 - stored PL/SQL in database, 10-51
- virtual hosts
 - See* multiports
- virtual paths
 - configuring, 9-19
 - creating, 3-17

W

- web browser
 - displaying database browser information, 10-50
- web configuration form, 8-23
- Web Request Broker
 - See* WRB, 2-1
- web site, 1-6, 2-6
 - adding nodes, 2-7
 - level, 2-1
 - navigation tree, 1-6
 - management, 2-1
- website40 site, 2-1
- Welcome page, 1-4
 - accessing, 1-5

- WRB, 2-6
 - administration, 2-9
- wrberr.html, 8-25

X

- XLF logging, 10-14, 10-29